DOCUMENT RESUME

ED 327 138 IR 014 756

AUTHOR Reck, Gene
TITLE Keeping Pace with New Technology: An Introduction to Robotics, FORTH, and Artificial Intelligence.
PUB DATE Dec 86
NOTE 124p.; Master's Practicum, Nova University.
PUB TYPE Dissertations/Theses - Practicum Papers (043) -- Computer Programs (101)

EDRS PRICE MF01/PC06 Plus Postage.
DESCRIPTORS Artificial Intelligence; *Computer Science Education; Computer Software; Electronic Equipment; *Instructional Development; Needs Assessment; Postsecondary Education; Pretests Posttests; *Programing Languages; *Robotics; Two Year Colleges

ABSTRACT

A course was developed to introduce students at a community college to four major areas of emphasis in emerging technologies: FORTH programming language, elementary electronic theory, robotics, and artificial intelligence. After a needs assessment indicated the importance of such a course, a pretest focusing on the four areas was given to students enrolled in the computer science curriculum. During the 16-week semester, students received instruction and completed class projects, including the design and construction of a robot. A posttest revealed an average increase of more than 10 percentage points in each of the four areas, thus indicating successful achievement of the stated outcome objectives. Appendices include testing instruments, test results, robot design diagrams, enhancements included in the fig-FORTH programming language, the course syllabus, programming assignments, and support materials. (18 references) (DB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED327138

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

KEEPING PACE WITH NEW TECHNOLOGY:

AN INTRODUCTION TO ROBOTICS, FORTH, AND ARTIFICIAL INTELLIGENCE

by

Gene Reck

A Practicum Report
submitted to the Faculty of the Center for the Advancement of Education
of Nova University in partial fulfillment of the requirements for the
degree of Master of Science.

The abstract of this report may be placed in the School
Practices Information Files for reference.

December/1986

Running Head: ROBOTICS

IR 014 452

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Gene Reck

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

I hereby testify that this paper and the work it reports are entirely my own. Where it has been necessary to draw from the work of others, published or unpublished, I have acknowledged such work in accordance with accepted scholarly and editorial practice. I give this testimony freely, out of respect for the scholarship of other workers in the field and in the hope that my own work, presented here, will earn similar respect.

Keeping Pace With New Technology: An Introduction to Robotics, FORTH, and Artificial Intelligence.
Reck, Eugene C., 1986: Practicum Report, Nova University, Center for the Advancement of Education
Descriptors: Robots/Robotics/Robotology/FORTH/ Artificial Intelligence/Expert Systems

A model course intended to introduce college students to emerging technologies was developed by the author. The goal of this course was to educate students in areas that will prove to be of vital importance in the future. The pace of technological change often outstips the evolution of curricula in educational institutions. This pilot course indicated one of several possible directions that computer science education may lead during the next few years.

The course featured four major areas of emphasis: FORTH programming, elementary electronic theory, robotics, and artificial intelligence. A test focusing on these areas was given to students enrolled in the prescribed computer science curriculum. A similar test was given as a pretest to students who enrolled in the course. Upon completion of the pilot course, students were given a similar examination as a posttest. An average increase of at least ten percentage points in each of the four areas emphasized by this course was the stated outcome objective. (Appendices include testing instruments, test results, robot design diagrams, enhancements included in the fig-FORTH programming language, a syllabus of the course, programming assignments, and support materials.)

		PAGE
TITLE PAGE		i
AUTHORSHIP STATEMENT		ii
ABSTRACT		iii
TABLE OF CONTENTS		iv
OBSERVER'S VERIFICATION		v
CHAPTER		
I	PURPOSE	1
II	RESEARCH AND SOLUTION STRATEGY	4
III	METHOD	11
IV	RESULTS	18
V	RECOMMENDATIONS	22
REFERENCE LIST		24
APPENDICES		
APPENDIX A:	TESTING INSTRUMENT	26
APPENDIX B:	TEST RESULTS	36
APPENDIX C:	ROBOT ASSEMBLY INSTRUCTIONS	40
APPENDIX D:	fig-FORTH ENHANCEMENTS	57
APPENDIX E:	COURSE SYLLABUS	68
APPENDIX F:	PROGRAMMING ASSIGNMENTS	73
APPENDIX G:	SUPPORT MATERIALS	102

KEEPING PACE WITH NEW TECHNOLOGY: AN INTRODUCTION TO ROBOTICS, FORTH, AND ARTIFICIAL INTELLIGENCE

I. PURPOSE

This document describes a situation within a Community College environment. The subjects of this study were college students interested in computers. The students could have been computer science majors, engineering majors, computer electronics majors, or any other type of student interested in this particular topic. The college has a well equipped microcomputer lab, comprised mainly of 64K Apple II+ computers.

The Computer Science curriculum consists of courses in a wide variety of programming languages. The introductory course in data processing and the computer literacy course include BASIC programming and use of application software such as word processors, databases, and spreadsheets. The introductory course in computer science includes Pascal programming. COBOL programming, advanced COBOL programming, FORTRAN programming, RPG programming, and Assembly language programming are also offered as part of the curricula. The Computer Electronics department offers a diverse curricula concerning electronic theory and practice.

The author of this document is presently the Supervisor

of the microcomputer lab, as well as an adjunct instructor. Decisions regarding the curricula usually result from group discussions among faculty members, with final approval dependent upon the Division Chairman. An instructor thus has limited power to determine curriculum changes.

"The use of robots and computer-aided design and manufacturing systems throughout the industries of the United States will be an increasing factor in the maintenance of our industrial leadership in the world." (Dorf, 1983) "Our obligation in the community college is to address these areas and to be on the leading edge of instruction in technology." (Lieff, 1986) An examination was given to students enrolled in Computer Science courses (See Appendix A). This examination was intended to sample student knowledge in four contemporary areas of Computer Science: FORTH programming, robotics, electronics, and artificial intelligence. These topics are currently presented in the introductory courses, but are not covered in detail within the existing curriculum. Low scores on this examination (See Appendix B) supported the position that the current curricula is neglecting to educate students in at least four important areas concerning the emerging technology. Students given the examination received an

average score of 56 percent correct on the subject of robotics, 49 percent correct on the subject of electronics, 20 percent correct on the subject of FORTH programming, and 25 percent correct on the subject of artificial intelligence. None of these scores represented acceptable levels of understanding of the topics under consideration. Such low scores indicated a need for expanding the current Computer Science curriculum to include coverage of these emerging technologies.

The major thrust of this project was to significantly enhance the knowledge and understanding of students enrolled in the course in four fields within the broader context of numerous emerging technologies. The four areas chosen for this study included: FORTH programming, robotics, electronics, and artificial intelligence. Over a period of one sixteen week semester, the students in the target group were to improve their knowledge of FORTH programming by at least ten percentage points as measured by the differences in scores between the pretest and posttest. Over the same period of one sixteen week semester, the students in the target group were to improve their understanding of robotics by at least ten percentage points as measured by the differences in scores between the pretest and posttest. Over the same period of one sixteen week

semester, the students in the target group were to improve their knowledge of electronics by at least ten percentage points as measured by the differences in scores between the pretest and posttest. Over the same period of one sixteen week semester, the students in the target group were to improve their understanding of artificial intelligence principles by at least ten percentage points as measured by the differences in scores between the pretest and posttest.

II. RESEARCH AND SOLUTION STRATEGY

"FORTH has been in use for over ten years in critical scientific and industrial applications." (Brodie,1981) "When better robots are built, they'll most likely be programmed in FORTH, a language created by Charles Moore in the early 1970s. FORTH is already the most popular language for industrial robots, and it may become the language of choice for personal robots as well." (Blankenhorn, 1985) "One of FORTH's strong points is in controlling equipment, whether machine or electronic." (Reymann,1983) "One of FORTH's major benefits is that it is a fast-programming, high-level language which remains close to machine code in efficiency and therefore in execution speed. While its words are human

language oriented, its procedures are closer to machine procedural, and it executes very quickly." (Reymann, 1983) "Overall, FORTH is one of the fastest means of producing efficient, compact, tested programs." (Reymann,1983) "One of the major advantages of FORTH is that the programmer is more in control of the computer and its resources than in most other high-level languages." (Reymann,1983) "FORTH has often been characterized as offbeat, totally unlike any other popular language in structure or in philosophy. On the contrary, FORTH incorporates many principles now boasted by the most contemporary languages. Structured design, modularity, and information-hiding are among the buzzwords of the day." (Brodie,1984) "The most attractive feature of FORTH is that it can be extended. That is, if you need to perform some function that is not already included in the language, you can add it!" (Scanlon,1982) One of the major focal points of this course was the introduction of the student to programming in a language known as FORTH. FORTH was chosen as the programming language for this course due to several reasons. First, FORTH is a major language in the areas of process control and robotics. It provides the programmer direct access to the hardware with which he is operating. Second, FORTH is extensible; that is, the vocabulary of the language can be permanently expanded. Thus

FORTH is able to "learn". This made it especially suitable to a course dealing with artificial intelligence. Third, FORTH is interactive; it provides immediate feedback to the programmer. Interactive languages tend to be "friendlier" and easier to learn. Fourth, FORTH is very compact and fast; making it an ideal language to use on a microcomputer where memory space is limited. The fifth reason for choosing FORTH is economics. Many versions of FORTH are available in the public domain, which is to say free. Thus FORTH is a good language for educators and schools that are usually forced to operate on a limited budget. The text that was used in this course was Starting FORTH by Leo Brodie (1981). This book is the standard introductory FORTH manual.

Another major area of this course was the study of robotics. The number of robots used by industry has increased dramatically in recent years. In 1982, there were approximately 15,000 first-generation industrial robots in use throughout the world. "It is estimated that the robot population will continue to expand at a rate of some 35 per cent through the 1980s." (Warring, 1984) In deciding the type of robot used as the focal point of this project, a variety of options were considered. Several commercially available robot kits were examined. Most of these were found

to be either electronic toys or prohibitively expensive. To qualify as a true robot, any device would have to conform to the accepted definition of a robot, which would be "an electro-mechanical device which when programmed can perform tasks, manipulation, or locomotion under automatic control". (Robotics, 1985) The device would have to be capable of performing three basic functions: input, processing, and output. None of the toys examined met the requirement that the machine be programmable. The programmable machines were all prohibitively expensive. It was thus decided to design a machine that would meet the requirements at an acceptable cost. (see Appendix C for construction details) The robot would be controlled by a microcomputer. The decision to construct a robot for use in this project followed the reasoning of Mr. Clive Hornbeck who suggested that the purchase of a prepackaged system would be unwise "because when you're on the job, you will likely be working with uniquely configured robotic workcells. You have to know how to build one from scratch ("Robotics Curriculum",1986)."

Thus students in the class were expected to build and program such a device, following plans provided by the instructor. (Schultz,1986) .Robots are an important social and economic consideration. Each year robots displace thousands of workers from their jobs. Managers

and administrators must be able to evaluate their needs in order to see whether robots are economically advantageous or detrimental. "Several factors need to be overcome to effect more rapid diffusion of automated manufacturing technologies in the U.S. These include lack of knowledge about robots and flexible manufacturing, inadequate appreciation for their potential, and, especially, the short-term investment horizon held by most American managers." (Dorf, 1983) A great deal of myth and fiction surround robots. By studying the principles upon which robotics are based, students received an opportunity to explore a topic of computer science that is still in its infancy.

In order to understand the operation of this educational robot, a basic understanding of electronic principles and components was necessary. Thus, another major objective of this course was to introduce students to basic electronic principles. Any number of computer controlled devices could have been constructed, but robots were chosen due to their increasingly important role in industry and manufacturing. "Robots can be found spray painting, welding, forging, loading/unloading heavy objects and doing extremely precise or delicate tasks." (Robotics, 1985) No computer science course at our institution dealt

with the problem of interfacing hardware devices to the computer. The computer science curriculum also did little to educate students on the electrical theories that make microcomputers possible. Many students have little or no experience performing simple electronic construction or repair. Students interested in computer science should understand how to make computers work in conjunction with other devices. "It has become apparent that there is a certain group of individuals who would like to use their machines as extensively as is possible, both at home and at their place of employment. This inevitably results in the need for interface circuitry to allow the computer to be connected to external devices allowing users to monitor and control their environment." (Hallgren, 1982)

The final component of this course, artificial intelligence is an area currently under rapid development and increasing research. "Much of the recent interest in this field has come in response to the Japanese Fifth Generation computer project, an ambitious effort whose goal is the creation of a new class of computing machinery that will make our present computers look like Stone-Age artifacts." (Thornburg, 1986) Artificial intelligence may be defined as "a machine's ability to mimic or improve on the

human ability to comprehend and reason". (Rosenthal, 1986) In dealing with the topic of artificial intelligence, we closely followed those issues presented by Graham. Graham pointed out several areas of current research in the area of artificial intelligence. These areas included "problem solving, natural language processing, perception and pattern recognition, information storage and retrieval, control of robots, game playing, automatic programming, and computational logic". (Graham, 1979) Robotics and artificial intelligence are related concepts since future robots "will require less input from the user and will rely more heavily on artificial intelligence and sensory input to determine required motions". (Dorf, 1983) Students were asked to develop programs for their robots that navigated a variety of mazes. "Whenever two or more computer buffs get together and begin discussing robots and artificial intelligence the subject of maze solving seems to crop up". (Heiserman, 1981) According to Heiserman, "games are the best way to challenge a machine's aptitude, the program, and yourself". (1981)

This course was designed to educate students on topics that are on the cutting edge of emerging technologies. There were no hard and fast solutions. There were few precedents to follow.

III. METHOD

The proposed course took place over a period of a single sixteen week semester. The students were expected to have completed at least one introductory course in the computer science curriculum and have had at least some exposure to computer programming concepts. The class met for three hours per week and counted as a three semester hour computer science elective for those students enrolled. A pretest (See Appendix A) was given to the students enrolled in the course during the first class session. Students were provided with enhanced versions of FORTH (See Appendix D) to use in the completion of their programming objectives. Apple and IBM PC versions of FORTH were available, but the Apple version was designated as the standard because the microcomputer lab was equipped with Apple II+ computers. Plans had been acquired to permit students to construct their own computer-controlled robot, complete with sensors, motors, and steering capability (See Appendix C). A midterm examination was utilized to provide feedback on the progress of the class eight weeks into the semester. (See Appendix G) The midterm exam was

worth 70 points of the students' total grade. The posttest was the students' final examination. (See Appendix A) The final examination was worth 100 points of the students' total grade. There were 13 programming projects worth ten points each for a total of 130 points. Thus, a student had the possibility of earning as many as 300 points during the semester. Grades for the course were determined by the number of points earned by the students. The grading scale was based on percentages, with 90 to 100 percent being an A, 80 to 89 percent being a B, 70 to 79 percent being a C, 60 to 69 percent being a D, and less than 60 percent being an F.

The course was organized around the four major components of robotics, FORTH programming, electronics, and artificial intelligence. Each component composed a module that interrelated to the other modules of the course. The individual components were not all of the same duration and it was necessary to deal with several of the topics simultaneously. It was intended that all the objectives of the course be integrated into a common, centralized series of student projects.

The course began with the module concerning FORTH programming. This module started with the first class session and ran for the duration of the sixteen week course.

This module closely followed the content of the textbook chosen for the course. Students were expected to read one chapter per week from the text and do the assigned programming projects for that chapter. The programming projects could be written either on the Apple II+ computers in the microcomputer lab, or on the students' home computer if they had access to the appropriate software. Versions of FORTH were provided for Apple compatible computers and for computers running IBM PC compatible MS-DOS. The programming projects were adopted from the text. These class projects were assigned with the intention that they should assist the students in their attempt to master the subject matter (See Appendix F) Lectures supplemented the reading, with particular attention being given to pointing out the differences between the text and the version of FORTH being used by the students. The programming projects were collected, reviewed in class, and graded to form part of the basis of the students' grade for the course. Each programming project was worth a maximum of ten points. This provided weekly feedback on the activities of the students. The first twelve weeks centered around the textbook with emphasis on instructing the students in FORTH programming fundamentals and techniques. The last four weeks required the students to apply the FORTH programming skills they had

acquired during the earlier twelve weeks. During the final four weeks, the students were presented with programming assignments that required them to program a computer controlled robot to solve various maze problems. The final four weeks represented a synthesis of all the modules presented in the course, where robotics, FORTH programming, electronics, and artificial intelligence were all tied together into a single cohesive whole. (See Appendix B for a complete course syllabus with a weekly breakdown of the topics presented. See Figure 1 for a timeline of the course.)

The electronics module began on the second class session and lasted for a period of five weeks. During this module students were required to follow a detailed set of plans and participate in the construction of a computer controlled robot. This robot was capable of being interfaced to Apple II microcomputers and programmed in FORTH. The class was divided into two groups of seven individuals per group. Each student was provided with a complete set of plans, including a discussion of the robot's theory of operation. Each group was provided with the necessary parts and materials to construct a fully functional robot. A fully assembled robot was provided as a

Timeline for Robotics Course

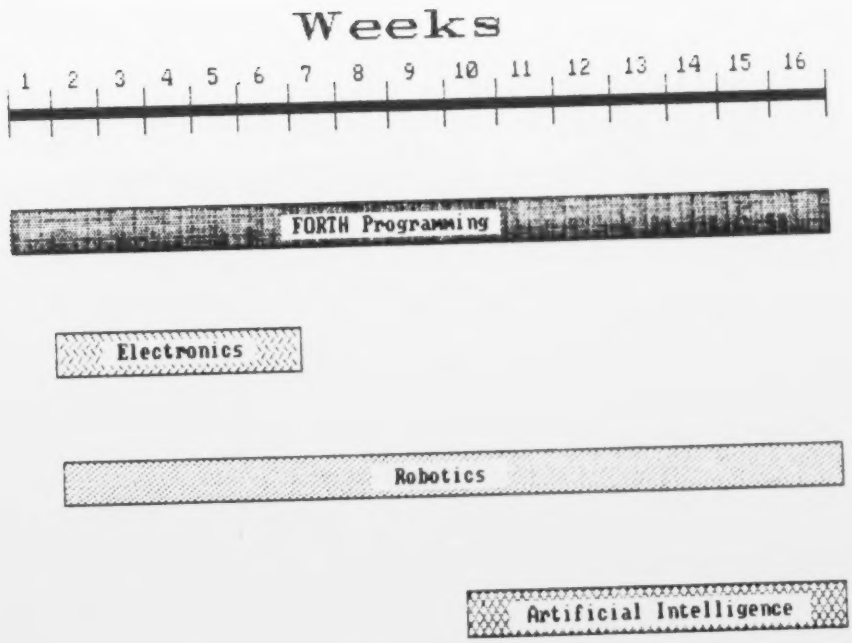


Figure 1

model for the students to view. The necessary tools and supplies were provided for the assembly operation. One and one-half hours per week were devoted to the construction of the robots during this five week period. The first session presented a discussion of fundamental electronic principles and terms, as well as an explanation of the robot construction project. The groups were assigned by the instructor. The next three sessions provided the students with time to construct the robots. The construction was supervised by the instructor, and the instructor was available to interpret the construction plans if necessary. The last session was devoted to testing the operation of the fully assembled robots. Test routines were written by the instructor in FORTH to test all of the various functions of the robots. Students were required to trace and correct any assembly mistakes. Once completed, the robots were set aside until the final four weeks of the semester. (See Appendix C for a complete discussion of the robot assembly instructions.)

The robotics component of the course began with the second class session and lasted for the entire sixteen weeks. Emphasis was placed on the industrial applications of robots, but home and other uses were also discussed. A

large number of filmstrips and video tapes were presented to familiarize students with material beyond the scope of the classroom environment. Industrial applications of robots were presented, with discussions on cost/benefit analysis, human labor displacement, technological obsolescence, and other contemporary issues concerning robotics. A number of contemporary magazine and journal articles were presented to the students. Lectures supplemented the textual and audio-visual materials. Students were familiarized with the historical development of robots. The class was required to become familiar with the terminology needed in any discussion dealing with robots. The students were acquainted with the various classification schemes of industrial robots, and the characteristics of members of each category. An educational robot trainer was introduced to the students. The Heath Educational Robot (HERO 1) was utilized as a means for demonstrating many of the principles upon which industrial robots are based. Students were made aware of the technology that made robots possible, and the directions research is taking to improve that technology.

The artificial intelligence component of the course began during the tenth week and lasted for the remaining six weeks of the course. This module consisted largely of discussion concerning artificial intelligence techniques,

historical developments in artificial intelligence, current research, and future trends. Several areas of artificial intelligence research were discussed, including expert systems, natural language processing, perception and pattern recognition, control of robots, automatic programming, problem solving, and game playing. A number of programs demonstrating artificial intelligence concepts were provided by the instructor. The students were required to interact with many of these demonstration programs. The final two projects required the students to write programs for their robots that attempted to impart a degree of artificial intelligence to the behavior of the robot.

IV. RESULTS

The scores on the final examination were analyzed and an average score was determined for each of the four areas covered by this course. These average scores were compared with the average scores obtained by students on the pretest and in the needs assessment study. (See Figure 2 and Appendix B.)

It was stated as an outcome objective that an overall average increase of at least ten percentage points

TEST RESULTS COMPARISON

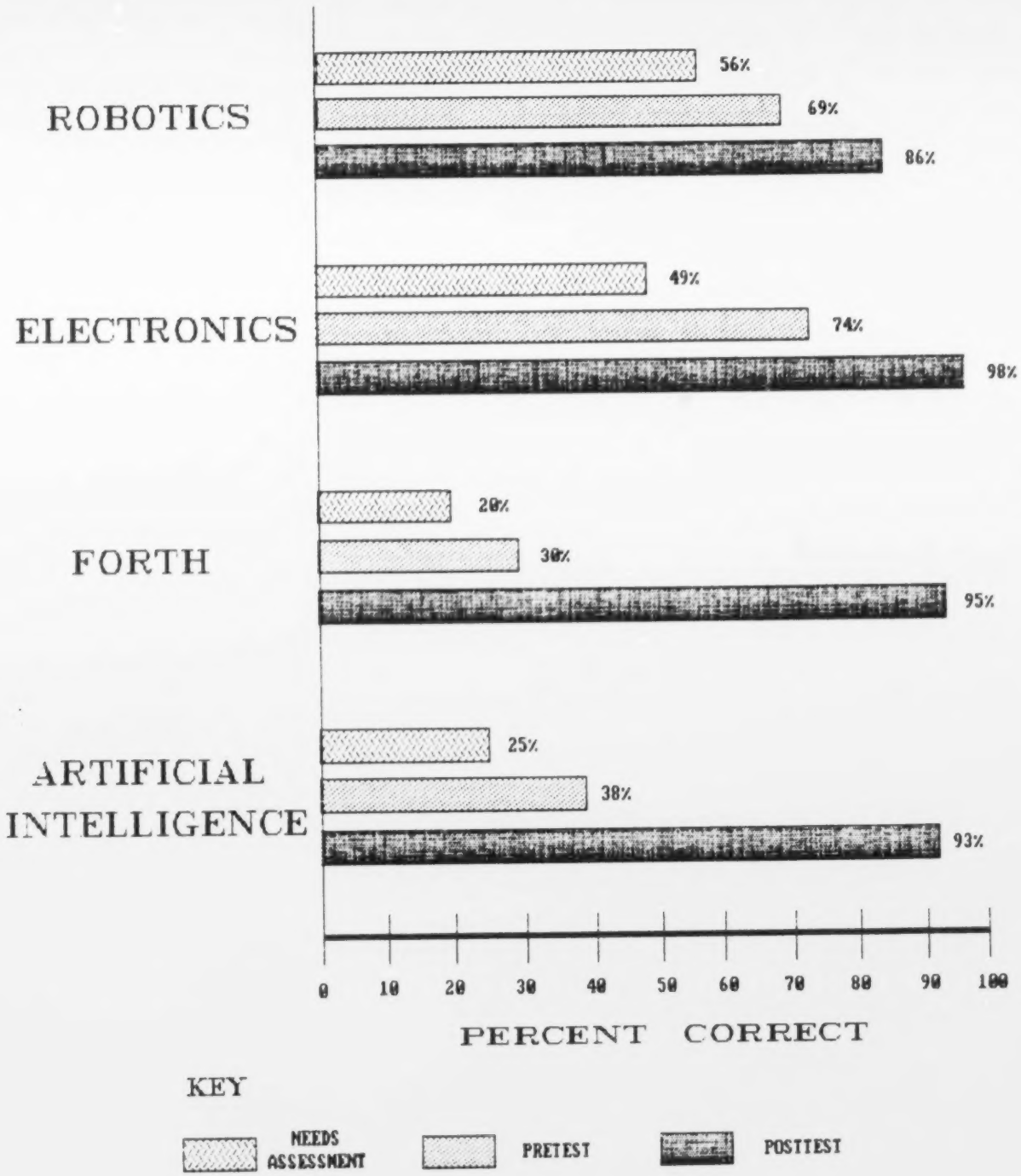


Figure 2

in the area of FORTH programming between the pretest and the posttest would be considered satisfactory. Students answered 20 percent of the questions correctly in the area of FORTH programming during the needs assessment study. Students correctly answered 30 percent of the questions concerning FORTH programming on the pretest. At the conclusion of the course, students correctly answered 95 percent of the questions concerning FORTH programming. Thus, there was an increase of 65 percent from the pretest to the posttest in the area of FORTH programming. Therefore the outcome objective concerning FORTH programming was successfully achieved.

It was also stated as an outcome objective that an overall average increase of at least ten percentage points in the area of robotics between the pretest and the posttest would be considered acceptable. Students answered 56 percent of the questions correctly in the area of robotics during the needs assessment study. Students correctly answered 69 percent of the questions concerning robotics on the pretest. At the conclusion of the course, students correctly answered 86 percent of the questions concerning robotics on the posttest. Thus, there was an increase of 17 percent from the pretest to the posttest in the area of

robotics. Therefore the outcome objective concerning robotics was also successfully accomplished.

Another outcome objective stated that an overall average increase of at least ten percentage points in the area of electronics between the pretest and the posttest would be considered satisfactory. Students answered 49 percent of the questions correctly in the area of electronics during the needs assessment study. Students correctly answered 74 percent of the questions concerning electronics on the pretest. At the conclusion of the course, students correctly answered 98 percent of the questions concerning electronics on the posttest. Thus, there was an increase of 24 percent from the pretest to the posttest in the area of electronics. Therefore the outcome objective concerning electronics was also successfully achieved.

The final outcome objective stated that an overall average increase of at least ten percentage points in the area of artificial intelligence between the pretest and the posttest would be considered acceptable. Students answered 25 percent of the questions correctly in the area of artificial intelligence during the needs assessment study. Students correctly answered 38 percent of the questions concerning artificial intelligence on the pretest. At the

conclusion of the course, students correctly answered 93 percent of the questions concerning artificial intelligence on the posttest. Thus, there was an increase of 55 percent from the pretest to the posttest in the area of artificial intelligence. Therefore the outcome objective concerning artificial intelligence was successfully fulfilled.

V. RECOMMENDATIONS

The four areas chosen for this particular study were but a few of the many new, emerging technologies. A continuing series of courses related to emerging technologies could be developed. A few other topics that could easily be developed into other Special Topic courses include telecommunications, interactive video, laser disk technology, graphic overlay techniques, CAD/CAM, CAI, and office automation. As technology continues to advance, the subject matter of these Special Topic courses would necessarily change. Some may even be incorporated into the regular academic curriculum. Whatever the status of the technology, a series of courses dealing with emerging technologies would always appeal to those individuals who wish to keep abreast of their time.

In addition, parts of this course could be separated from the whole and used to implement variations on the theme. The inexpensive robot built by the students could be programmed in other languages, such as BASIC, independent of any study of artificial intelligence. Likewise, FORTH programming could be taught without any consideration of either robotics or artificial intelligence. And artificial intelligence could be discussed while only briefly mentioning either robotics or FORTH. Thus, some of the components may prove themselves useful in a variety of other settings.

There is a great deal of material to introduce in a course of this nature. Since the subject matter deals with emerging technologies, there are few static issues involved. The topics under consideration are still under development and subject to rapid, sweeping changes. Extensive use should be made of contemporary journal and magazine articles. Use of audio-visual materials, such as videotapes and filmstrips, is also encouraged. In this way, a course such as this one may change as developments in these areas make it necessary to do so.

BIBLIOGRAPHY

Apple II fig-FORTH Assembly Source Listing . (1981) . San Carlos, CA: FORTH Interest Group.

Blackenhorn, Dana . (1985, January) . Personal Robots: The PC on Wheels . PC World, pp. 177 - 184.

Brodie, Leo . (1981) . Starting FORTH . Englewood Cliffs, NJ: Prentice-Hall, Inc.

Brodie, Leo . (1984) . Thinking FORTH . Englewood Cliffs, NJ: Prentice-Hall, Inc.

Dorf, Richard C. (1983) . Robotics and Automated Manufacturing . Reston, VA: Reston Publishing Company, Inc.

fig-FORTH Installation Manual . (1980) . San Carlos, CA: FORTH Interest Group.

Hallgren, Richard C. (1982) . Interfacing Projects for the Apple II . Englewood Cliffs, NJ: Prentice-Hall, Inc., pp. 1 - 2.

Hartnell, Tim . (1985) . Exploring Artificial Intelligence on Your Apple II . New York, NY: Bantam Books.

Heiserman, David L. (1981) . Robot Intelligence with experiments . Blue Ridge Summit, PA: TAB Books Inc.

Lieff, M., Gonzalez, X. F., Stoskus, J., Ayres, M. C. (Spring 1986) . Community College Transition Into The Information Age . Technical Education News, p. 5.

Reymann, Joseph . (1983) . Understanding FORTH . Sherman Oaks, CA: Alfred Publishing Co, Inc.

Robotics . (1985) . Chatsworth, CA: Career Aids, Inc.

Robotics Curriculum at Iowa Community College Emphasizes the Basics . (1986, January) . T.H.E. Journal, p. 74.

Rosenthal, Steve . (1986, March) . Applewords . A+, p.61.

Scanlon, Leo J. (1982) . FORTH Programming . Indianapolis: Howard W. Sams & Co., Inc.

Schultz, Charles P. (1986) PIR-1 and PIRPL a project in robotics education, Barry University, Miami, FL.

Thornburg, David D. (1986, March) . Artificial Intelligence . A+, pp. 55 - 59.

Warring, R. H. (1984) . Robots & Robotology . Blue Ridge Summit, PA: TAB Books Inc.

APPENDIX A

TESTING INSTRUMENT

AND

ANSWER KEY

1. Which nation leads the world in production and use of robots?
   A. Germany
   B. Russia
   C. Japan
   D. America

2. The use of a machine or group of machines to carry out a process under the direction of a human operator is known as:
   A. Networking
   B. Automation
   C. Decentralization
   D. Mechanization

3. The performance of a specific combination of actions by a machine or group of machines "automatically" - without the aid of a human operator is known as:
   A. Networking
   B. Automation
   C. Telecommunication
   D. Mechanization

4. A general-purpose programmable machine that can do any of several tasks under stored program control is known as:
   A. Modem
   B. Robot
   C. Multiplexor
   D. Heuristic

5. The part of the robot that performs the actual work is called the
   A. Controller
   B. Manipulator
   C. Rotating drum
   D. Servo control system

6. Spray painting and welding operations are best handled by which type of robot?
   A. Medium-technology
   B. Limited-sequence
   C. Low-technology
   D. High-technology

7. Which category of robot is best suited for machine loading/unloading of medium weight objects?
   A. Low-technology
   B. Puma series 250
   C. Medium-technology
   D. High-technology

8. Which device helped make the special purpose robot economically practical?
   A. Transistor
   B. Microprocessor
   C. Computer
   D. Hydraulic cylinder

9. What is considered the brains and nervous system of the robot?
   A. Manipulator
   B. Controller
   C. Actuator
   D. Gripper

10. Low-technology robots are best suited for which type of task?
   A. Welding operations
   B. Palletizing material
   C. Material handling
   D. Forging

11. The electronic symbol means:

   A. Capacitor
   B. Transistor
   C. Resistor
   D. Diode

12. An electromechanical device in which contacts are opened and/or closed by variations in the conditions of an electric circuit thereby affecting the operation of other devices in the same or other circuits is known as a(n)
   A. Relay
   B. Transformer
   C. Transistor
   D. Microprocessor

13. The Apple family of microcomputers uses which type of microprocessor?
   A. 6809
   B. 6502
   C. 8088
   D. Z80

14. The type of computer memory that is permanent and cannot be changed is known as:
    A. RAM
    B. REM
    C. RPM
    D. ROM

15. The type of computer memory that goes blank when the computer's power is turned off is known as:
    A. RAM
    B. REM
    C. RPM
    D. ROM

16. A type of computer chip known for its low heat production and low power consumption is:
    A. RAM
    B. ROM
    C. CMOS
    D. MOS

17. The electronic schematic symbol means:

    A. Inductor
    B. Ground
    C. Transformer
    D. Antenna

18. A bit represents which of the following:
    A. 1 or 0
    B. A binary digit
    C. An on or an off
    D. All of the above

19. The collating sequence used by most microcomputers to represent characters is known as:
    A. EBCDIC
    B. NECC
    C. ASCII
    D. FICC

20. The phrase 16K refers to exactly how many bytes?
    A. 16,000
    B. 16,192
    C. 16,256
    D. 16,384

21. The primary means by which information is shared between the different components of a Forth program is through use of:
    A. Variables
    B. Files
    C. Records
    D. Stack

22. The fundamental unit of a Forth program is:
    A. The procedure
    B. The word
    C. The byte
    D. The function

23. The most common way to create a new definition in Forth is through use of a:
    A. Define-New-Instruction
    B. Procedure declaration
    C. Colon definition
    D. Begin/End block

24. Which of the following is NOT a characteristic of Forth?
    A. Fast
    B. Compact
    C. Extensible
    D. Extensive Error Checking

25. Which of the following is the Forth instruction that fetches a byte from memory?
    A. Get
    B. C@
    C. F>
    D. B^

26. A double length number in Forth is how long?
    A. 8 bits
    B. 16 bits
    C. 32 bits
    D. 64 bits

27. Which of the following is a Forth instruction that stores a byte into memory?
    A. C!
    B. B<
    C. C@
    D. B#

28. Which of the following instructions is used to eliminate a Forth program or definition?
    A. NEW
    B. FORGET
    C. ERASE
    D. DELETE

29. Forth makes use of which type of arithmetic notation?
    A. Prefix
    B. Infix
    C. Postfix
    D. Suffix

30. Which of the following is NOT a typical characteristic of Forth?
    A. Extensive Error Checking
    B. Fast Execution Speed
    C. Extensibility
    D. Interactive

31. Which of the following number systems is Forth able to operate in?
    A. Decimal
    B. Hex
    C. Binary
    D. All of the above

32. Forth definitions are found in what type of structure?
    A. File
    B. Record
    C. Dictionary
    D. Stack

33. Which of the following is NOT a Forth loop structure?
    A. DO - LOOP
    B. LOOP - ENDLOOP
    C. BEGIN - UNTIL
    D. BEGIN - WHILE - REPEAT

34. Which of the following is an invalid name for a Forth definition?
    A. &H
    B. 1A
    C. HELLO
    D. None of the above

35. Which of the following symbolizes the end of a Forth definition?
    A. ;
    B. END
    C. .
    D. /*

36. The Forth data stack obeys which of the following principles?
    A. FIFO
    B. FILO
    C. LIFO
    D. LILO

37. Which of the following Forth commands does NOT affect the stack?
    A. SWAP
    B. DUP
    C. +
    D. None of the above

38. Which of the following Forth commands will print out a single-length number from the stack?
    A. WRITE
    B. PRINT
    C. .
    D. #

39. Which of the following is NOT a standard Forth mathematical operator?
    A. X
    B. /
    C. +
    D. None of the above

40. Which of the following is used to begin a Forth definition?
    A. BEGIN
    B. :
    C. Define-New-instruction
    D. DEF

41. Which of the following is not a major area of interest to researchers working in AI?
    A. Computer Literacy
    B. Pattern Recognition
    C. Control of Robots
    D. Natural Language Processing

42. Which of the following programs is an example of an Expert System?
    A. HAL
    B. Eliza
    C. Caduceus
    D. Ractor

43. Any hint or rule of thumb that helps us discover the solution to a problem is known as a(n):
    A. Mnemonic
    B. Heuristic
    C. Algorithm
    D. Anomaly

44. Which of the following is a widely known program which imitates human intelligence?
    A. Shazam
    B. Abraham
    C. Xemu
    D. Eliza

45. Which of the following types of searches could be used to solve a maze problem?
    A. Breadth-first
    B. Depth-first
    C. Ordered
    D. All of the above

46. Which of the following is the major disadvantage to Blind Searching?
    A. Coding Complexity
    B. Logical Anomalies
    C. Combinatorial Explosion
    D. Algorithmic Extraction

47. A proverb, such as "Look before you leap", is an example of a(n):
    A. Heuristic
    B. Dynamic variable
    C. Pattern
    D. Automatic program

48. The classic artificial intelligence programming language is:
    A. BASIC
    B. COBOL
    C. LISP
    D. Forth

49. The test of artificial intelligence where two terminals were placed in a room, one connected to a computer and the other to another terminal where a human was sitting, and the test subject was asked to identify which was which was known as:
    A. The AI Test
    B. The Turing Test
    C. The Ferguson Test
    D. The Jenson Test

50. What has greatly contributed to the exposive growth of AI research in the last few years?
    A. Japan's Fifth Generation Computer Project
    B. America's world lead in industrial robotics
    C. Russia's intense interest in the microcomputer industry
    D. The explosive growth in information on how the human brain functions

ANSWER KEY FOR TESTING INSTRUMENT

1.	C
2.	D
3.	B
4.	B
5.	B
6.	D
7.	C
8.	B
9.	B
10.	C
11.	C
12.	A
13.	B
14.	D
15.	A
16.	C
17.	B
18.	D
19.	C
20.	D
21.	D
22.	B
23.	C
24.	D
25.	B
26.	C
27.	A
28.	B
29.	C
30.	A
31.	D
32.	C
33.	B
34.	D
35.	A
36.	C
37.	D
38.	C
39.	A
40.	B
41.	A
42.	C
43.	B
44.	D
45.	D
46.	C
47.	A
48.	C
49.	B
50.	A

APPENDIX B

TEST RESULTS

TEST RESULTS INDICATING NEED

CLASS	ROBOTICS	ELECT.	FORTH	A.I.
		(% CORRECT)		
COC 1110	50	20	15	20
COC 1110	40	60	25	10
COC 1110	70	60	30	40
COC 1110	60	80	25	20
COC 1110	40	40	20	50
COC 1110	80	40	15	20
COC 1110	70	50	10	0
COC 1110	80	40	15	40
COC 1110	20	20	20	50
COC 1110	50	20	30	20
COC 1110	30	30	30	20
COC 1110	60	50	30	30
COC 1110	50	40	25	20
COC 1110	70	30	30	30
COC 1110	40	60	0	0
COC 1110	60	60	20	20
COC 1110	40	30	20	40
COC 1110	60	30	10	10
COC 1110	60	40	15	50
COC 1110	60	50	5	20
COC 1110	60	60	15	20
COC 1110	50	60	25	30
COC 1110	80	80	45	20
COC 1110	50	60	40	30
COC 1110	60	50	5	10
COC 1110	70	80	15	30
COC 1110	70	50	15	30
COC 1110	70	70	15	10
COC 1110	30	30	10	30
COC 1300	70	70	5	20
COC 1300	30	50	40	30
AVG. PERCENT CORRECT	56	49	20	25

RESULTS OF PRETEST

STUDENT	ROBOTICS	ELECT.	FORTH	A.I.
		(% CORRECT)		
Monte	70	70	40	10
Matt	80	70	15	30
Scott	80	80	20	50
Dan	70	70	20	40
Ken	70	90	45	50
Ted	80	90	25	50
Jim	60	70	70	10
Tony	80	90	30	40
Sandy	60	70	30	30
Pam	40	60	20	40
Sue	80	70	20	60
Dot	70	60	20	40
Mark	80	80	40	30
Greg	50	60	25	50
AVG. PERCENT CORRECT	69	74	30	38

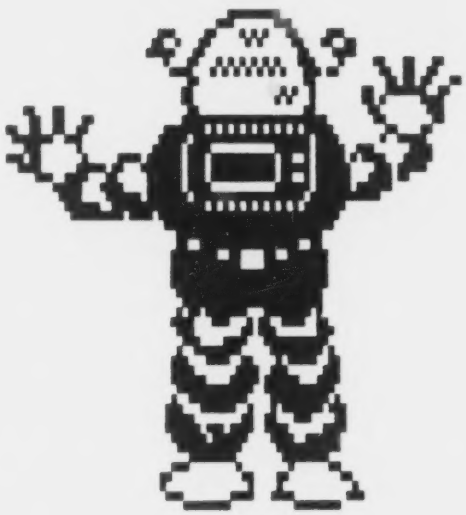
RESULTS OF POSTTEST

STUDENT	ROBOTICS	ELECT. (% CORRECT)	FORTH	A.I.
Monte	100	100	80	80
Matt	80	100	95	100
Scott	80	100	100	80
Dan	70	90	90	100
Ken	80	100	95	100
Jim	90	100	95	70
Tony	100	100	100	100
Sandy	100	100	100	100
Pam	70	90	90	90
Sue	100	100	100	100
Greg	80	100	100	100
AVG. PERCENT CORRECT	86	98	95	93

APPENDIX C

ROBOT ASSEMBLY INSTRUCTIONS

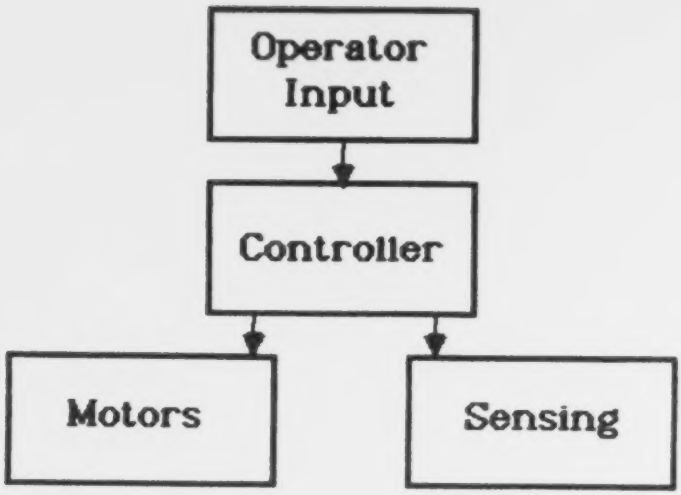
Robot Construction Project



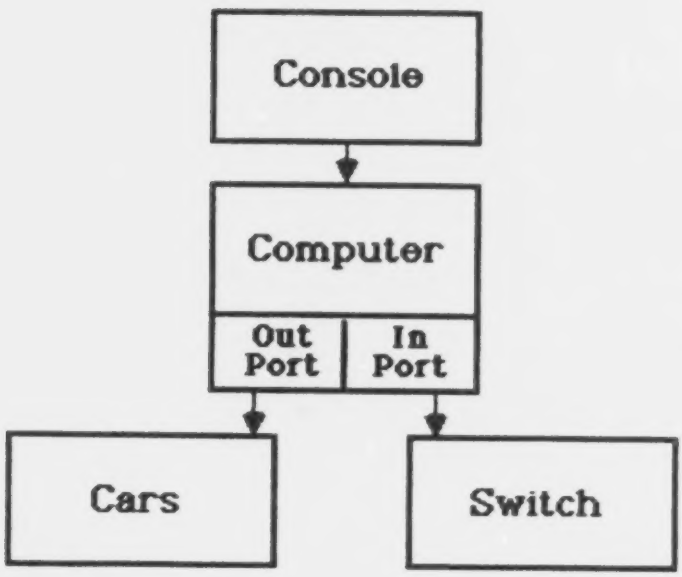
CAP 2930

Gene Reck

Robotic Systems



Typical arrangement of a robotic system



Arrangement of our educational robotic system

ROBOT THEORY OF OPERATION

The robot chosen for construction in class has been designed to use two inexpensive battery powered cars that can be controlled by a personal computer through four relays, which can cause either car to go forward or backward as desired.

A switch mounted on the front of the robot acts as a sensing mechanism which allows detection of contact between the robot and an object in front of it, thus permitting artificial intelligence programming.

The robot has been designed with a connector that permits the robot to be interfaced with a variety of different types of computers.

A chip (74LS367) is used to boost the power coming from the computer to a level where it is sufficient to activate the relays. The relays are single-pole, double-throw (SPDT) reed relays that are activated by a +5 volt signal, thus they can be switched by computer logic level signals since most microcomputers operate internally at 5 volts.

The robot moves by activating the relays connected to the car motors. The relay contact acts as a switch, applying the battery voltage to one side of the motor, causing it to move the car. A signal set to logic level 1 (approximately +5 volts) closes the contact. If the contact on the plus side of the motor is closed, the car will go forward. If the negative side contact is closed, the car will go backward.

The cars are mounted in opposite directions for added stability. To make the robot go forward or backward, the cars must go in opposite directions. To turn the robot, the cars must go in the same direction.

The contact sensing switch is attached to a signal that is normally at logic level 1 by being connected to +5 volts through a 10K ohm resistor. When contact is made between the switch and some solid object, the signal is connected through the switch to ground, which is logic level 0.

Though admittedly crude, this robot demonstrates many of the principles characterizing all robotic systems. The basic design may be modified or expanded as students gain knowledge and confidence in the principles involved.

ROBOT COMPONENTS

The Robot has been designed so that the only assembly skills required are soldering, glueing, wire preparation, and using a hobby knife. The parts used in the robot are relatively few, inexpensive, and easily obtainable.

The following is a partial list of vendors who carry some of the parts used in the construction of the robot:

All Electronics Corporation
905 S. Vermont Avenue
P.O. Box 20406
Los Angeles, CA 90006
1-800-826-5432

DIGI-KEY Corporation
701 Brooks Ave. South
P.O. Box 677
Thief River Falls, MN 56701
1-800-344-4539

Hi Tek Sales
119r Foster Street
Peabody, MA 01961-3357
(617) 532-2323

JAMECO Electronics
1355 Shoreway Road
Belmont, CA 94002
(415) 592-8097

K-Mart
contact local store

Radio Shack
contact local store

ROBOT PARTS LIST

Part Description	Qty	Vendor	Part #	Price
Race Car	2	K-Mart	6252	$ 3.97 ea.
		Hi Tek	85D0264	$ 1.00 ea.
				$ 5.00 /6
AA Battery	2	K-Mart		$ 2.99 /4
		Hi Tek	85V0360	$ 0.69 ea.
				$ 5.00 /10
		Radio Shk	23-552	$ 2.99 /4
			23-125	$ 3.99 /4
Battery Holder	2	Radio Shk	270-401	$ 0.39 ea.
5 Volt SPDT Reed Relays	4	DIGI-KEY	Z619-ND	$ 4.40 ea.
				$36.90 /10
		All Elect	RRLY-DIP5	$ 1.75 ea.
				$16.00 /10
74LS367	1	DIGI-KEY	74LS367ND	$ 0.68 ea.
		All Elect	74LS367	$ 0.65 ea.
		JAMECO	74LS367	$ 0.59 ea.
Female DB-25 Connector	1	Radio Shk	276-1548	$ 3.99 ea.
		DIGI-KEY	1S-25S-ND	$ 3.16 ea.
				$25.87 /10
		All Elect	DB25S	$ 2.95 ea.
				$25.00 /10
		JAMECO	DB25S	$ 2.95 ea.
Switch	1	Radio Shk	275-016	$ 1.19 ea.
		All Elect	SMS-350	$ 0.75 ea.
10K Ohm resistor	1	Radio Shk	271-1335	$ 0.39 /5
		DIGI-KEY	10KQ	$ 0.25 /5
16 Pin DIP Socket	1	Radio Shk	276-1998	$ 0.89 /2
Stiff Cardboard (5"x6" min)				

INTERFACE CABLE PARTS LIST

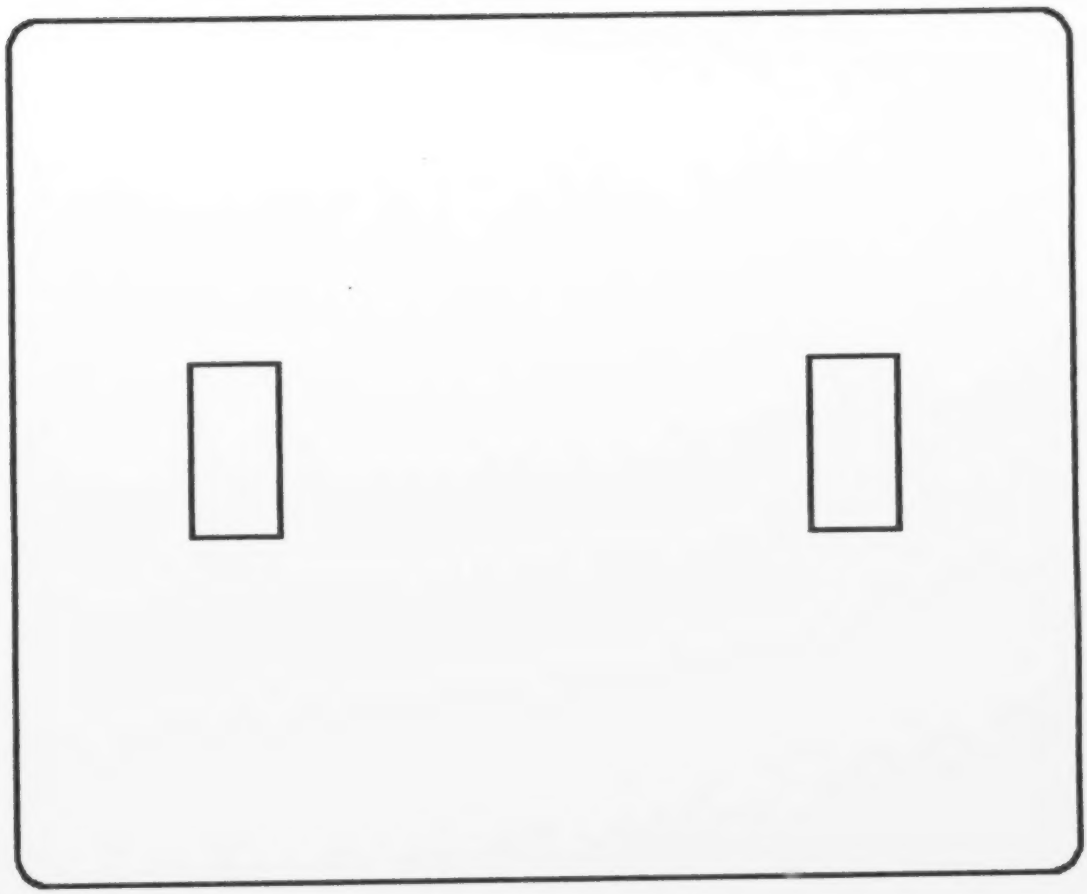
Part Description	Qty	Vendor	Part #	Price
25 Conductor Ribbon Cable	5'	Radio Shk	278-772	$ 3.59 /5'
	10'	Hi Tek	4BS0594	$ 6.00 /10'
				$20.00 /50'
Male DB-25 Connector	1	Radio Shk	276-1547	$ 2.99 ea.
		Hi Tek	85BS0149	$ 1.25 ea.
				$10.00 /9
		DIGI-KEY	1S-25P-ND	$ 2.42 ea.
				$19.80 /10
16 pin Male DIP Header	1	Radio Shk		
		DIGI-KEY	R8103-ND	$ 0.92 ea.
				$ 8.14 /10
		All Elect	16HIC	$ 0.55 ea.
		JAMECO	16PINHP	$ 0.69 ea.
				$ 6.50 /10

CONSTRUCTION MATERIALS LIST

Hobby Knife
Pin or Needle
All-Purpose Adhesive
Wire Cutter
Wire Stripper
Needle-nose pliers
Soldering Iron
Roll of electrical solder
Electrical tape
Ruler
Felt tip marker (fine)
Pencil

Figure 1

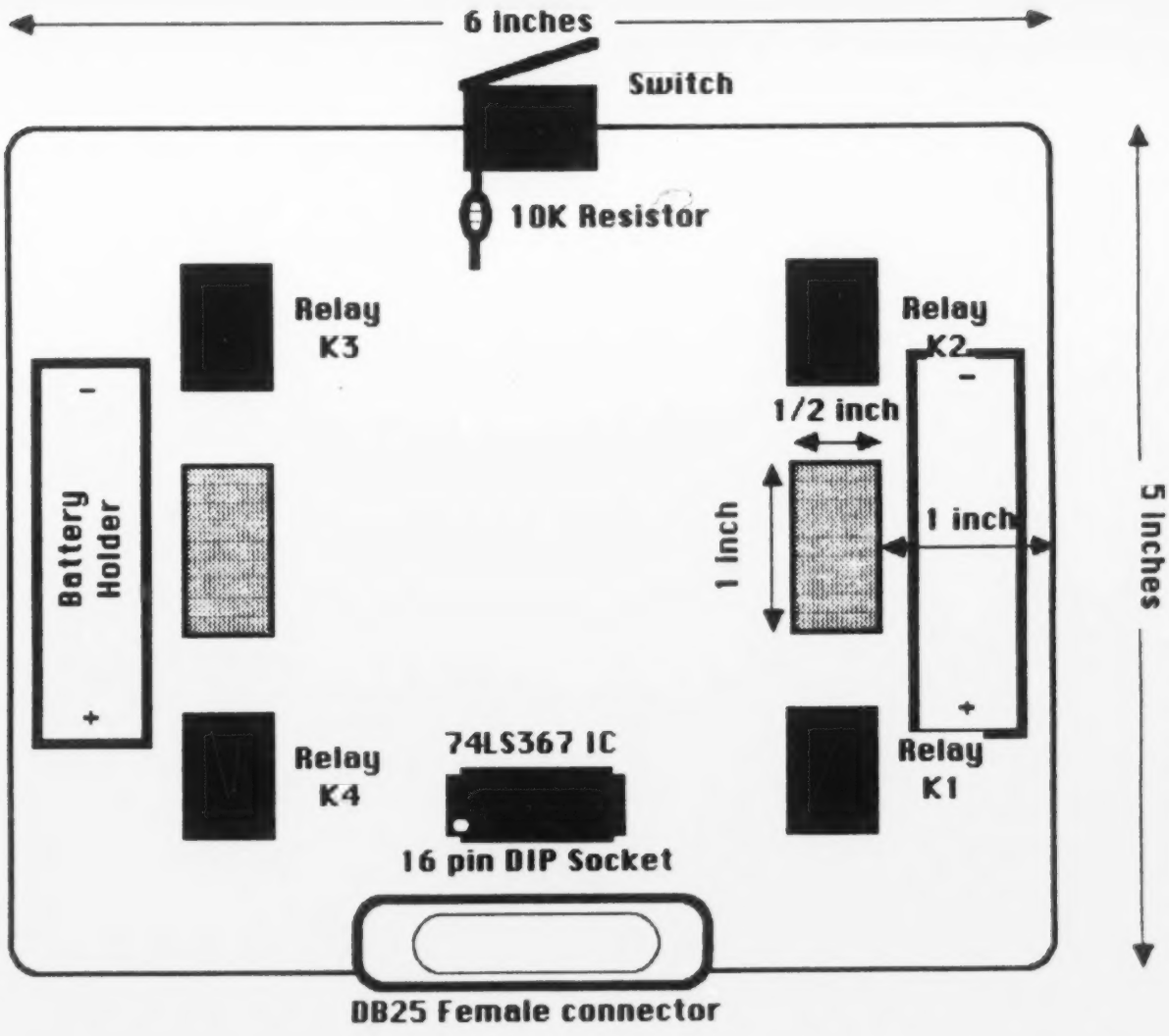
Robot base template



This is the template for the base of the robot. Trace this pattern onto a sheet of sturdy cardboard and cut out the base using a sharp hobby knife.

Figure 2

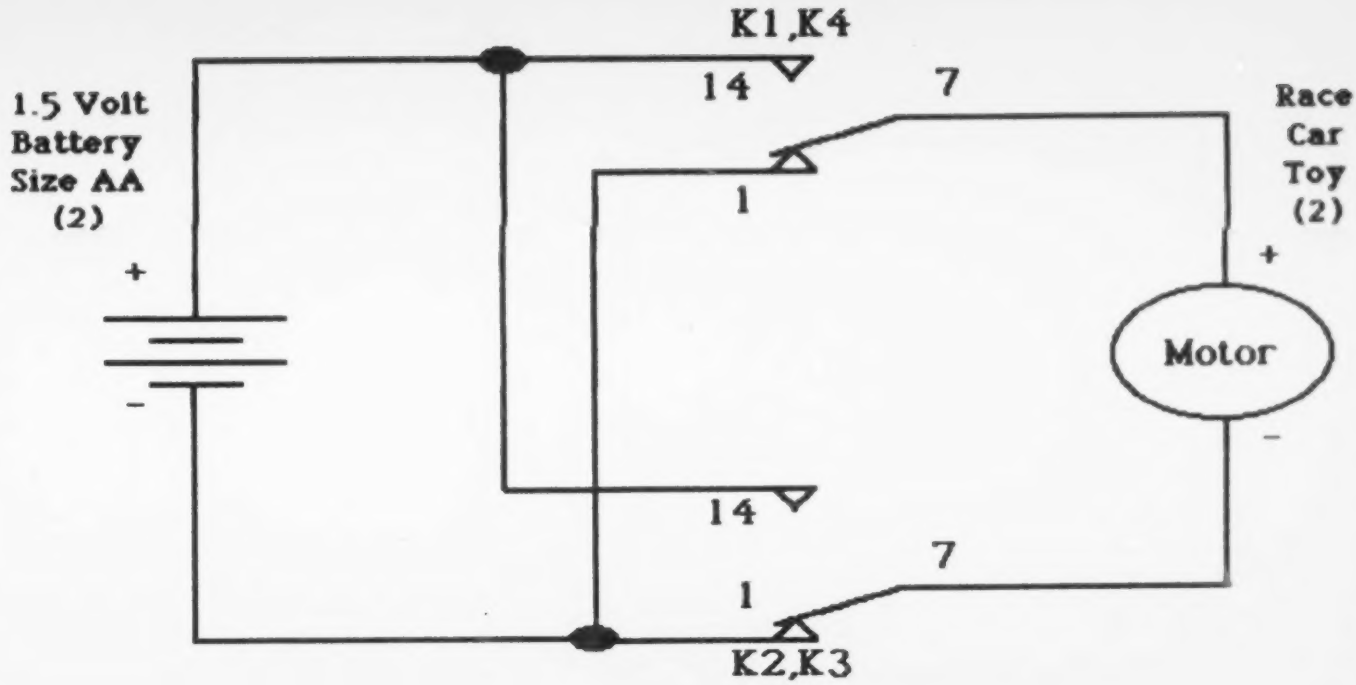
Robot component layout



Mount the components to the cardboard base as shown in the diagram above. Use a pin or a needle to make holes for the electrical pins to fit through. For the 16 pin DIP socket, cut two parallel slits with a sharp hobby knife. Also use the knife to cut a notch for the DB25 connector. Use glue sparingly to secure the components to the cardboard.

Figure 3

Robot Schematic Diagrams



K1,K2,K3,K4 = +5VDC SPDT Reed Relays

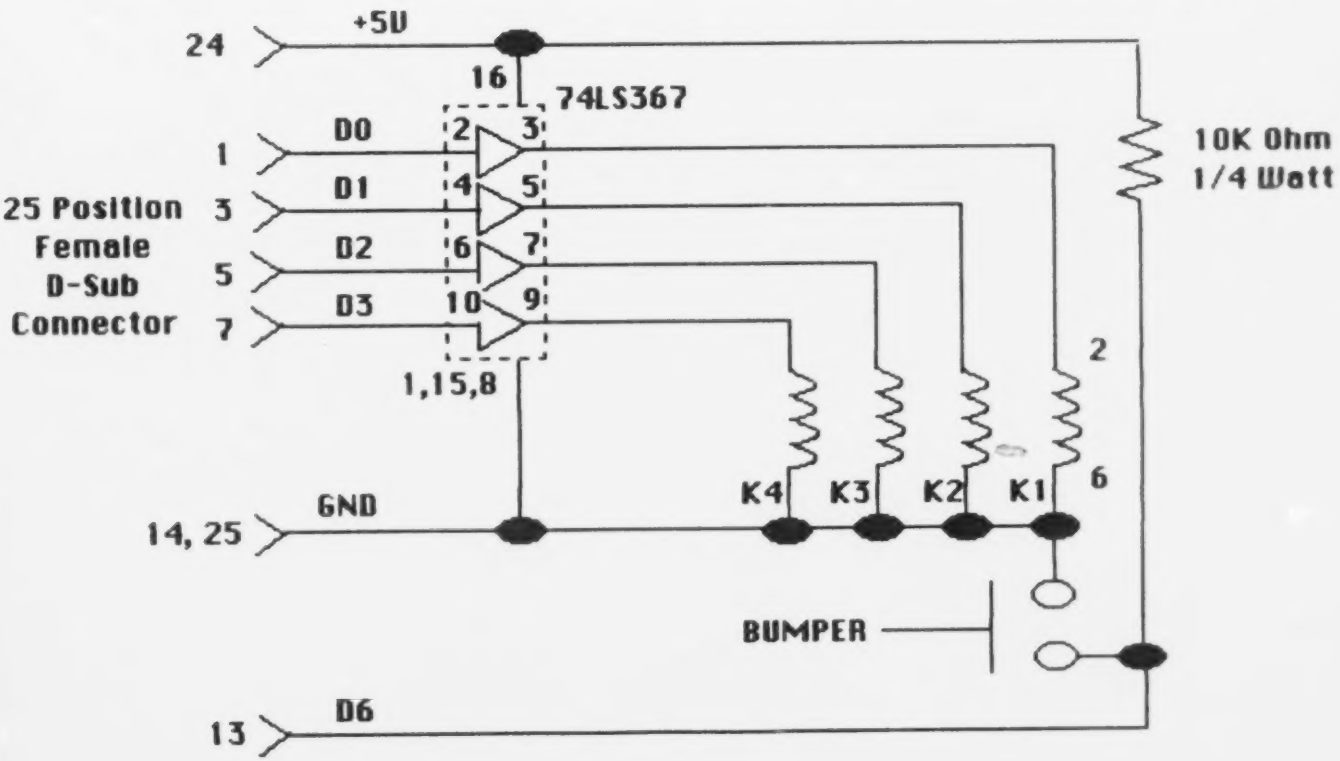
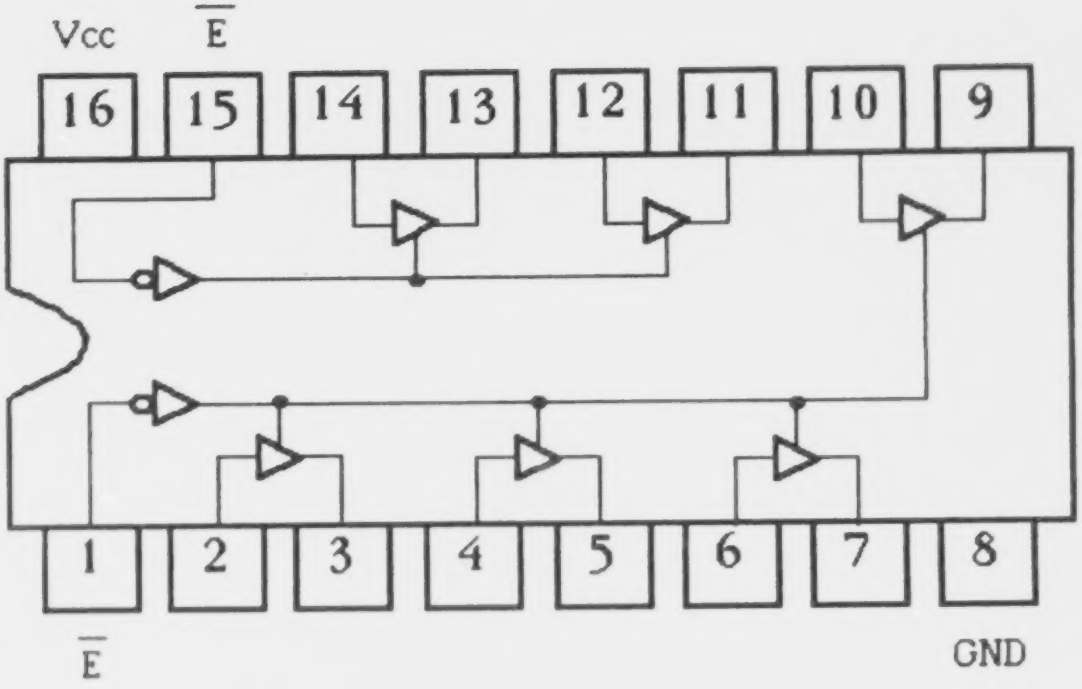


Figure 4

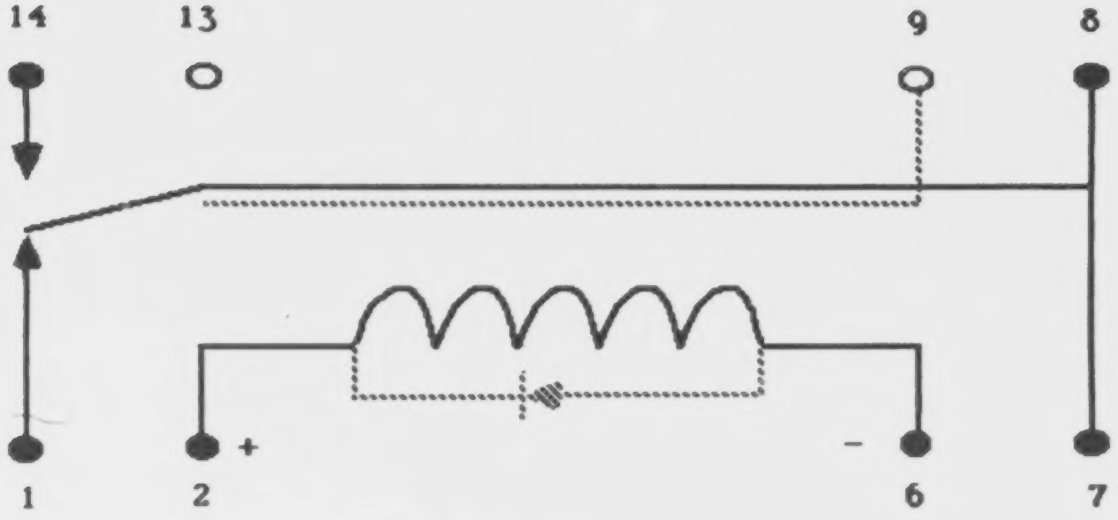
74LS367 IC

Tri-state Hex Buffer



This chip boosts the low current signals coming from the computer to a level where they can serve to activate the relays. It essentially performs the role of a type of amplifier, allowing the computer to control the robot.

5V DC SPDT
DIP REED RELAY



This diagram represents the pin-outs for the reed relays we will be using in construction of the robot.

Note

If you use different relays, you will need to consult an electronics reference book to find the correct pin-outs unless they are provided by the manufacturer.

This is probably the single, most difficult step in the construction of the robot.

*** Be Careful ***

ROBOT ASSEMBLY INSTRUCTIONS

1) Cut out the cardboard base using the template provided or design one of your own. (See Figure 1) Paint and decorate now if you wish to do so.

2) Mount battery holders, relays, DIP socket, and roller switch to base using an all-purpose adhesive. (See Figure 2)

3) Separate the 25 conductor ribbon cable into 16 and 9 conductor lengths. The component wiring for the robot will be taken from the 9 conductor length. The 16 conductor length will be used for the interface cable.

4) Prepare four 4" lengths of wire. Cut four 4" conductors from the 9 conductor cable. Separate the four conductors into four separate wires. Bare both ends of each wire for about 1/4 inch. Solder the 4" lengths of wire to the + and - terminals of each toy car.

5) Solder wires from the DB-25 female connector to the pins of the DIP socket. (See Figure 3)

DB-25	16 pin DIP Socket
24	16
1	2
3	4
5	6
7	10
25	1
25	15
25	8

6) Connect the outputs from the pins of the 16 pin DIP socket to the relays. (See Figure 3)

16 pin DIP Socket	Coil on Relay
3	K1
5	K2
7	K3
9	K4

7) Connect the other side of the coil of each relay to pin 14 of the female DB-25 connector. (4 wires)

8) Connect the 10K Ohm resistor and pin 13 of the DB-25 to the common lead of the switch.

9) Connect +5V from pin 24 of the DB-25 to the other end of the resistor.

10) Connect the normally open side of the switch to pin 25 of the DB-25 connector (GND).

11) Bring the car wires through the base cut-outs, and place the base on the cars. Don't glue the base to the cars yet! Note that the car connected to relays K1 and K2 should be facing the front of the base, while the other car should be facing the opposite direction.

12) Connect the wires from the cars to the relays. (See Figure 3)

13) Connect the batteries to the relays. (See Figure 3) Connect + to normally open of K1, K2, K3, and K4. Connect - to normally closed of K1, K2, K3, and K4.

14) Cut a notch in the cardboard base for the DB-25 female connector. Mount the DB-25 connector to the robot using the adhesive.

15) Plug the 74LS367 chip into the 16 pin DIP socket. Make sure the pins of the chip correspond to the wiring of the DIP socket.

16) Use electrical tape to insulate any wiring connections that may cause shorts.

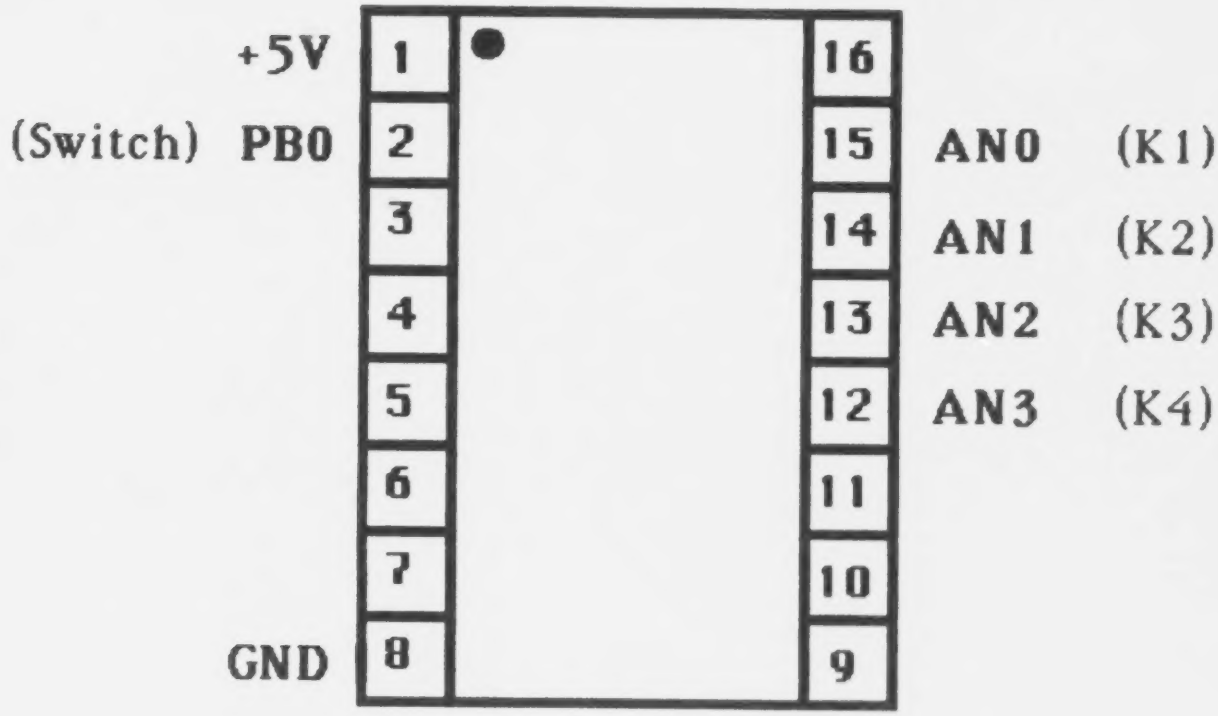
17) Assemble the interface cable according to the information to follow.

18) Be sure to check for shorts between battery terminals and from +5V to GND. If any exist, correct the wiring mistakes. When the wiring is correct, plug in the robot's batteries, connect the robot to the computer, and you're ready to GO!

19) Once you have determined that the robot functions as it is supposed to, use the all-purpose adhesive to glue the cars to the cardboard base.

Figure 5

APPLE II GAME I/O CONNECTOR



ADDRESS:

AN0	Off	= 49240
	On	= 49241
AN1	Off	= 49242
	On	= 49243
AN2	Off	= 49244
	On	= 49245
AN3	Off	= 49246
	On	= 49247
PB0		= 49249

PIN 1 - +5V Power Supply
PIN 2 - Single Bit Input
PIN 8 - System Electrical Ground
PINS 12 thru 15 - Annunciator Outputs

INTERFACE DESIGN

Interfacing the robot to a computer usually only requires the basic computer configuration. Many different types of computers will be able to control this robot, not just Apple II computers. It is simply a matter of redesigning the interface cable to allow a different type of computer to communicate with the robot.

The Apple II interface uses the Game I/O connector on the motherboard. (See Figure 5) Only four motors can be controlled because only four output bits are available. The output bits are controlled by "soft switches". Referencing one address turns the switch on, referencing another address turns the switch off.

To use the Game I/O connector for output, each bit is turned on or off by referencing a different address for each bit. In FORTH you can use the C! instruction to reference the appropriate address. In BASIC you would use the POKE instruction.

The contact switch is read by examining the particular location in memory assigned to that purpose. In FORTH you can use C@ to examine the proper address. In BASIC you would use the PEEK instruction.

To construct the Apple interface, you will connect a series of wires between the 16 pin DIP Header and the Male DB-25 connector. Use the 16 conductor cable you set aside earlier for the construction of this interface. Separate the 16 conductor cable into two 8 conductor sections. Bare both ends of all 8 conductors. Connect the conductors as follows:

DB-25	16 pin DIP Header
1	15
3	14
5	13
7	12
13	2
14	8
24	1
25	8

This completes the construction of the robot. Congratulations! Now the fun can begin.

```
SCR # 22
  0 ( ROBOT TEST INSTRUCTIONS )
  1
  2 49240 CONSTANT A0
  3 A0 2 + CONSTANT A1
  4 A1 2 + CONSTANT A2
  5 A2 2 + CONSTANT A3
  6
  7 : FORWARD
  8     0 A0 1 + C! 0 A1 C! 0 A2 1 + C!
  9     0 A3 C! ;
 10
 11 : BACKWARD
 12     0 A0 C! 0 A1 1 + C! 0 A2 C!
 13     0 A3 1 + C! ;
 14
 15 -->

SCR # 23
  0 ( ROBOT TEST INSTRUCTIONS CONT. )
  1
  2 : LEFT-TURN
  3     0 A0 C! 0 A1 1 + C! 0 A2 1 + C!
  4     0 A3 C! ;
  5
  6 : RIGHT-TURN
  7     0 A0 1 + C! 0 A1 C! 0 A2 C!
  8     0 A3 1 + C! ;
  9
 10 : STOP
 11     0 A0 C! 0 A1 C! 0 A2 C!
 12     0 A3 C! ;
 13
 14 : SENSE 49249 C@ ;
 15 ;S
```

APPENDIX D

fig-FORTH ENHANCEMENTS

```
SCR # 10
  0 ( TEXT, LINE, SQUARE BRACKET SUB. )
  1 FORTH DEFINITIONS   HEX
  2
  3 : EXEC 0 STATE ! ;  IMMEDIATE
  4
  5 : TEXT
  6    HERE C/L 1+ BLANKS WORD HERE PAD C/L 1+ CMOVE ;
  7
  8 : LINE
  9    DUP FFF0 AND 17 ?ERROR
 10    SCR @ (LINE) DROP ;
 11 -->
 12
 13
 14
 15

SCR # 11
  0 ( LINE EDITOR )
  1 VOCABULARY EDITOR IMMEDIATE   HEX
  2 : WHERE
  3    DUP B/SCR / DUP SCR ! ." SCR # " DECIMAL .
  4    SWAP C/L /MOD C/L * ROT BLOCK + CR C/L TYPE
  5    CR HERE C@ - SPACES 5E EMIT (COMPILE) EDITOR QUIT ;
  6
  7 EDITOR DEFINITIONS
  8 : #LOCATE
  9    R# @ C/L /MOD ;
 10 : #LEAD
 11    #LOCATE LINE SWAP ;
 12 : #LAG
 13    #LEAD DUP >R + C/L R> - ;
 14 : -MOVE
 15    LINE C/L CMOVE UPDATE ;  -->

SCR # 12
  0 ( LINE EDITING COMMANDS )
  1 : H
  2   LINE PAD 1+ C/L DUP PAD C! CMOVE ;
  3
  4 : E
  5   LINE C/L BLANKS UPDATE ;
  6
  7 : S
  8   DUP 1 - OE
  9   DO I LINE I 1+ -MOVE -1 +LOOP E ;
 10
 11 : D
 12   DUP H OF DUP ROT
 13   DO I 1+ LINE I -MOVE LOOP E ;
 14
 15 -->
```

```
SCR # 13
  0 ( LINE EDITING COMMANDS )
  1
  2 : M
  3   R# +! CR SPACE #LEAD TYPE 5F EMIT
  4   #LAG TYPE #LOCATE . DROP ;
  5
  6 : T
  7   DUP C/L * R# ! DUP H 0 M DROP ;
  8
  9 : L
 10   SCR @ LIST ;
 11  -->
 12
 13
 14
 15

SCR # 14
  0 ( LINE EDITING COMMANDS )
  1 : R
  2   PAD 1+ SWAP -MOVE ;
  3
  4 : P
  5   1 TEXT R ;
  6
  7 : I
  8   DUP S R ;
  9 CR
 10 : TOP
 11    0 R# ! ;
 12  -->
 13
 14
 15

SCR # 15
  0 ( SCREEN EDITING COMMANDS )
  1 : CLEAR
  2     SCR ! 10 0 DO FORTH I EDITOR E LOOP ;
  3
  4 : FLUSH
  5    EXEC LIMIT FIRST - B/BUF 4 + / ]
  6    LITERAL 0 DO 7FFF BUFFER DROP LOOP ;
  7
  8 : COPY
  9     B/SCR * OFFSET @ + SWAP B/SCR * B/SCR OVER + SWAP
 10    DO DUP FORTH I BLOCK 2 - ! 1+ UPDATE LOOP
 11    DROP FLUSH ;
 12  -->
 13
 14
 15
```

```
SCR # 16
  0 ( DOUBLE LENGTH OPERATORS )
  1
  2 FORTH DEFINITIONS
  3
  4 : 2DROP DROP DROP ;
  5
  6 : 2DUP OVER OVER ;
  7
  8 : 2SWAP ROT >R ROT R> ;
  9
 10 EDITOR DEFINITIONS
 11  -->
 12
 13
 14
 15

SCR # 17
  0 ( STRING MATCH FOR EDITOR )
  1 : -TEXT
  2     SWAP -DUP IF
  3       OVER + SWAP
  4       DO DUP C@ FORTH I C@ -
  5       IF 0= LEAVE ELSE 1+ THEN LOOP
  6       ELSE DROP 0= THEN ;
  7 : MATCH
  8    >R >R 2DUP R> R> 2SWAP OVER + SWAP
  9    DO 2DUP FORTH I -TEXT
 10    IF >R 2DROP R> - I SWAP - 0 SWAP 0 0 LEAVE
 11    THEN LOOP 2DROP
 12    SWAP 0= SWAP ;
 13
 14
 15  -->

SCR # 18
  0 ( STRING EDITING COMMANDS )
  1 : 1LINE
  2    #LAG PAD COUNT MATCH R# +! ;
  3
  4
  5 : FIND
  6    BEGIN 3FF R# @ <
  7      IF TOP PAD HERE C/L 1+ CMOVE 0 ERROR ENDIF
  8      1LINE UNTIL ;
  9
 10 : DELETE
 11    >R #LAG + FORTH R -
 12    #LAG R MINUS R# +!
 13    #LEAD + SWAP CMOVE
 14    R> BLANKS UPDATE ;
 15  -->
```

```
SCR # 19
  0 ( STRING EDITOR COMMANDS )
  1 : N
  2    FIND 0 M ;
  3
  4 : F
  5    1 TEXT N ;
  6
  7 : B
  8    PAD C@ MINUS M ;
  9
 10 : X
 11    1 TEXT FIND PAD C@ DELETE 0 M ;
 12
 13 : TILL
 14    #LEAD + 1 TEXT 1LINE 0= 0 ?ERROR
 15   #LEAD + SWAP - DELETE 0 M ;  -->

SCR # 20
  0 ( STRING EDITOR COMMANDS )
  1 : C
  2    1 TEXT PAD COUNT
  3    #LAG ROT OVER MIN >R
  4    FORTH R R# +!
  5    R - >R
  6    DUP HERE R CMOVE
  7    HERE #LEAD + R> CMOVE
  8    R> CMOVE UPDATE
  9    0 M ;
 10 FORTH DEFINITIONS   DECIMAL
 11 LATEST 12 +ORIGIN !
 12 HERE 28 +ORIGIN !
 13 HERE 30 +ORIGIN !
 14 ' EDITOR 6 + 32 +ORIGIN !
 15 HERE FENCE !      ;S

SCR # 30
  0 ( FORTH UTILITIES )
  1 DECIMAL
  2 : NOT 0= ;
  3
  4 0 VARIABLE RNDM     HERE RNDM !
  5 : RND RNDM @ 31421 * 6927 + DUP RNDM ! U* SWAP DROP ;
  6
  7 : ** DUP 1 = IF DROP ELSE DUP 0= IF DROP DROP 1 ELSE DUP 1 < IF
  8       DROP DROP 0 ELSE SWAP DUP ROT 1 DO OVER * LOOP SWAP DROP
  9       THEN THEN THEN ;
 10 -27258 CONSTANT S0
 11 : .S SP@ S0 @ - -DUP 0= IF CR ." STACK EMPTY " ELSE
 12       S0 @ DUP ROT + DO CR I ? 2 +LOOP THEN ;
 13 DECIMAL
 14 : BINARY 2 BASE ! ;
 15  -->
```

```
SCR # 31
  0 ( FORTH UTILITIES  CONTINUED )
  1
  2 : PICK 2 * SP@ + @ ;
  3
  4 : ROLL DUP 1 = IF DROP ELSE DUP 1 DO SWAP R> R> ROT >R >R >R
  5       LOOP 1 DO R> R> R> ROT ROT >R >R SWAP LOOP THEN ;
  6
  7 : J R> R> R> R SWAP >R SWAP >R SWAP >R ;
  8
  9 : I' R> R> R SWAP >R SWAP >R ;
 10
 11 : LEN 255 0 DO DUP I + C@ 0= IF I LEAVE THEN LOOP SWAP DROP ;
 12
 13 : ACCEPT OVER 1+ DUP ROT EXPECT LEN SWAP C! ;
 14
 15  -->

SCR # 32
  0 ( FORTH UTILITIES       CONTINUED )
  1
  2 : <CMOVE DUP ROT + SWAP ROT 1 - DUP ROT + DO 1 - I C@ OVER
  3           C! -1 +LOOP DROP ;
  4
  5 : CMOVE# OVER 4 PICK > IF <CMOVE ELSE CMOVE THEN ;
  6
  7 : -MATCH SWAP DUP C@ 5 PICK 5 ROLL + DUP 1 SWAP 6 PICK - 1+
  8       7 ROLL DO 3 PICK I C@ = IF 0 6 PICK 1 DO J I + C@ 6 PICK
  9       I + C@ = NOT IF DROP 1 LEAVE THEN LOOP IF ELSE DROP DROP
 10       I 4 PICK + 0 LEAVE THEN THEN LOOP ROT DROP ROT DROP
 11       ROT DROP ;
 12
 13 : -TEXT 0 ROT 0 DO DROP OVER I + C@ OVER I + C@ - DUP 0=
 14      NOT IF LEAVE THEN LOOP SWAP DROP SWAP DROP ;
 15  -->

SCR # 33
  0 ( DOUBLE LENGTH OPS )
  1
  2 : 2! DUP 2 + >R ! R> ! ;
  3
  4 : 2@ DUP 2 + @ SWAP @ ;
  5
  6 : 2OVER >R >R 2DUP R> R> 2SWAP ;
  7
  8 : 2ROT 6 ROLL 6 ROLL ;
  9
 10 : D- -1 D+- D+ ;
 11
 12 : D= D- 0= SWAP 0= AND ;
 13
 14 : D0= 0. D= ;
 15  -->
```

```
SCR # 34
  0 ( DOUBLE LENGTH OPS CONTINUED )
  1
  2 : D< D- SWAP DROP 0< ;
  3
  4 : DNEGATE DMINUS ;
  5
  6 : DMAX 2OVER 2OVER D< IF 2SWAP THEN 2DROP ;
  7
  8 : DMIN 2OVER 2OVER D< NOT IF 2SWAP THEN 2DROP ;
  9
 10 : 2, HERE 2! 4 ALLOT ;
 11
 12 : 2CONSTANT <BUILDS , , DOES> 2@ ;
 13
 14 : 2VARIABLE <BUILDS , , DOES> ;
 15 -->

SCR # 35
  0 ( APPLE-FORTH UTILITES  BY RECK )
  1
  2 : MYSELF LATEST PFA CFA , ;  IMMEDIATE
  3
  4 : ASCII BL WORD HERE 1+ C@ (COMPILE) LITERAL ;  IMMEDIATE
  5
  6 : DEPTH S0 @ SP@ - 2 - 2 / ;
  7
  8 HEX
  9 C030 CONSTANT SPEAKER
 10 DECIMAL
 11
 12 : SOUND 0 DO DUP 0 DO LOOP 0 SPEAKER C! LOOP DROP ;
 13
 14 : BELL 7 EMIT ;
 15 -->

SCR # 36
  0 ( MORE FORTH UTILITIES )
  1 8 CONSTANT DLEN
  2
  3 : DUMP
  4       CR HEX OVER + SWAP DO I 0 D.
  5       186 EMIT SPACE DLEN 0
  6       DO I J + C@ 3 .R LOOP CR
  7       7 SPACES DLEN 0
  8       DO I J + C@ DUP DUP 64 >
  9       SWAP 92 < AND IF
 10       SPACE 128 + EMIT SPACE ELSE
 11       DROP SPACE 174 EMIT SPACE
 12       ENDIF LOOP CR CR
 13       ?TERMINAL IF LEAVE ENDIF
 14       DLEN +LOOP CR ;
 15 -->
```

```
SCR # 37
  0 ( FORTH/MACHINE CODE INTERFACE )
  1 HEX
  2 : NEXT 4CEA , 6144 , ;
  3
  4
  5 : DRIVE DUP DUP 1 < SWAP 2 > OR IF .
  6         ." ? RANGE ERROR" CR ELSE
  7         B7EA C@ B7F8 C! B7EA C!
  8         ENDIF ;
  9
 10 : FLASH 7F 32 C! ;
 11 : INVERSE 3F 32 C! ;
 12 : NORMAL FF 32 C! ;
 13
 14 DECIMAL
 15 -->

SCR # 38
  0 ( MORE APPLE/FORTH UTILITIES )
  1 HEX
  2 : PR# DUP DUP 0 < SWAP 8 > OR IF
  3       SPACE . ." ? RANGE ERROR" CR
  4       ELSE DUP IF C0 + 37 C!
  5       00 36 C! ELSE FDF0 36 !
  6       DROP ENDIF ENDIF ;
  7
  8 DECIMAL
  9 ;S
 10
 11
 12
 13
 14
 15

SCR # 40
  0 ( APPLE/FORTH GRAPHICS BY RECK  )
  1 HEX
  2 : GMODE 0 C050 C! ;
  3 : TMODE 0 C051 C! ;
  4 : -MIX 0 C052 C! ;
  5 : MIX 0 C053 C! ;
  6 : PRIMARY 0 C054 C! ;
  7 : SECONDARY 0 C055 C! ;
  8 : LO-RES 0 C056 C! ;
  9 : HI-RES 0 C057 C! ;
 10 CREATE HGR 20EA , F3E2 , NEXT SMUDGE
 11 CREATE HGR2 20EA , F3DE , NEXT SMUDGE
 12 CREATE HCLR 20EA , F3F2 , NEXT SMUDGE
 13 CREATE BKGND 20EA , F3F6 , NEXT SMUDGE
 14 DECIMAL
 15 -->
```

```
SCR # 41
  0 ( APPLE/FORTH GRAPHICS BY RECK )
  1 HEX
  2 1C CONSTANT HCOLOR1
  3 6 CONSTANT HCOLOR
  4 CREATE HPLOT 488A , E0A5 , A548 ,
  5         48E1 , A5EA , 48E2 , E4A5 ,
  6         A548 , 48E5 , 06A5 , E485 ,
  7         03B5 , B5A8 , 4800 , 02B5 ,
  8         68AA , 5720 , 68F4 , E585 ,
  9         8568 , 68E4 , E285 , 8568 ,
 10         68E1 , E085 , AA68 , E8E8 ,
 11         E8E8 , NEXT SMUDGE
 12
 13 DECIMAL
 14 -->
 15

SCR # 42
  0 ( APPLE/FORTH LO-RES GRAPHICS )
  1 HEX
  2 CREATE CLRSCR 20EA , F832 , NEXT SMUDGE
  3 CREATE CLRTOP 20EA , F836 , NEXT SMUDGE
  4 CREATE NEXTCOL 20EA , F85F , NEXT SMUDGE
  5 CREATE SETGR 20EA , FB40 , NEXT SMUDGE
  6 CREATE SETCOL 00B5 , 20EA , F864 , E8E8 , NEXT SMUDGE
  7 CREATE PLOT 00B5 , B5A8 , 2002 ,
  8         F800 , E8E8 , E8E8 ,
  9         NEXT SMUDGE
 10 CREATE HLINE 00B5 , 2C85 , 02B5 ,
 11         B5A8 , 2004 , F819 , E8E8 ,
 12         E8E8 , E8E8 , NEXT SMUDGE
 13 DECIMAL
 14 -->
 15

SCR # 43
  0 ( APPLE/FORTH LO-RES GRAPHICS CON'T)
  1 HEX
  2 CREATE VLINE 00B5 , 2D85 , 04B5 ,
  3                B5A8 , 2002 , F828 ,
  4                E8E8 , E8E8 , E8E8 ,
  5                NEXT SMUDGE
  6 CREATE SCRN 00B5 , B5A8 , 2002 ,
  7               F871 , 0295 , 00A9 ,
  8               0395 , E8E8 , NEXT
  9               SMUDGE
 10 DECIMAL
 11 -->
 12
 13
 14
 15
```

```
SCR # 44
  0 ( HI-RES GRAPHICS CONTINUED )
  1 DECIMAL
  2 0 VARIABLE XCURRENT
  3 0 VARIABLE YCURRENT
  4 : POINT HCOLOR C! 2DUP YCURRENT !
  5          XCURRENT ! HPLOT ;
  6 : POS YCURRENT ! XCURRENT ! ;
  7 : RIGHT YCURRENT @ SWAP XCURRENT @
  8        DUP ROT DUP XCURRENT +! + 1+
  9        SWAP DO DUP I SWAP HPLOT
 10        LOOP DROP ;
 11 : LEFT YCURRENT @ SWAP XCURRENT @
 12        DUP ROT DUP MINUS XCURRENT
 13        +! - 1 - SWAP DO DUP I
 14        SWAP HPLOT -1 +LOOP DROP ;
 15 -->

SCR # 45
  0 ( HI-RES GRAPHICS CONTINUED )
  1 : DOWN XCURRENT @ SWAP YCURRENT @
  2       DUP ROT DUP YCURRENT +! + 1+
  3       SWAP DO DUP I HPLOT LOOP DROP ;
  4
  5 : UP XCURRENT @ SWAP YCURRENT @ DUP
  6       ROT DUP MINUS YCURRENT +! -
  7       1 - SWAP DO DUP I HPLOT
  8       -1 +LOOP DROP ;
  9 -->
 10
 11
 12
 13
 14
 15

SCR # 46
  0 ( JOYSTICK ROUTINES                )
  1 HEX
  2 CREATE PADDLE A88A , 00B5 , 98AA ,
  3                2048 , FB1E , AA68 ,
  4                00A9 , 0195 , 98EA ,
  5                0095 , NEXT SMUDGE
  6 : JOYSTICK 0 PADDLE 10 0 DO LOOP
  7            1 PADDLE 10 0 DO LOOP ;
  8 : OFIRE C061 C@ 7F > ;
  9 : 1FIRE C062 C@ 7F > ;
 10 : -FIRE OFIRE NOT 1FIRE NOT AND ;
 11 : BFIRE OFIRE 1FIRE AND ;
 12 DECIMAL
 13 -->
 14
 15
```

```
SCR # 47
  0 ( MORE HI-RES GRAPHICS                )
  1
  2 : SE XCURRENT @ YCURRENT @ ROT DUP
  3       DUP XCURRENT +! YCURRENT +!
  4       0 DO 2DUP I + SWAP I + SWAP
  5       HPLOT LOOP 2DROP ;
  6 : SW XCURRENT @ YCURRENT @ ROT DUP
  7       DUP MINUS XCURRENT +! YCURRENT
  8       +! 0 DO 2DUP I + SWAP I - SWAP
  9       HPLOT LOOP 2DROP ;
 10 : NW XCURRENT @ YCURRENT @ ROT DUP
 11       DUP MINUS XCURRENT +! MINUS
 12       YCURRENT +! 0 DO 2DUP I -
 13       SWAP I - SWAP HPLOT LOOP
 14       2DROP ;
 15 -->

SCR # 48
  0 ( MORE HI-RES GRAPHICS.CONTINUED )
  1
  2 : NE XCURRENT @ YCURRENT @ ROT DUP
  3       DUP XCURRENT +! MINUS YCURRENT
  4       +! 0 DO 2DUP I - SWAP I + SWAP
  5       HPLOT LOOP 2DROP ;
  6 ;S
  7
  8
  9
 10
 11
 12
 13
 14
 15
```

APPENDIX E

COURSE SYLLABUS

TERM: Fall, 1986

COURSE: Robotics, FORTH, and Artificial Intelligence

INSTRUCTOR: Gene Reck
Office - Room 415, Phone - ext. 2415
Office Hours - by appointment

PREREQUISITES: None

TEXTBOOK: Starting FORTH, by Leo Brodie

REQUIRED MATERIALS: Two 5.25 inch SS/DD floppy disks

GRADING POLICIES:

There will be 13 programming projects assigned, each worth 10 points. There will be two examinations, a midterm and a final. The midterm will be worth 70 points and the final will be worth 100 points. Thus, there will be a total of 300 points possible in this course.

GRADING SCALE:

270 - 300	A
240 - 269	B
210 - 239	C
180 - 209	D
Below 180	F

ATTENDANCE AND WITHDRAWALS:

Class attendance is mandatory. If you miss three classes you are subject to withdrawal for excessive absence. Withdrawals without penalty may be made during the first week of classes. Other withdrawals are at the instructor's discretion.

PLAGIARISM OR CHEATING:

Except in the case of group projects, all assignments are to be completed on an individual basis. Students found violating the spirit of this rule will be subject to disciplinary action in accordance with due process.

MAKE-UP EXAMS:

Students should make EVERY effort to take their exams at the specified dates and times. Should a problem arise where this is not possible, contact your instructor at the earliest opportunity to make arrangements for an alternate examination.

COURSE CONTENT:

This course will introduce students to the programming language known as FORTH. Principles of robotics will be studied and basic electronic skills will be developed. Students will have the opportunity to construct and program their own robot. Developments in the area of artificial intelligence will be explored.

COURSE SCHEDULE:

Week 1 : Introduction
Number Systems - Decimal, Binary, and Hexadecimal
Number Base Conversions
Read Chapter 1 in Starting FORTH
Assign Project 1
FORTH Fundamentals - the Dictionary and the Stack
Postfix Notation and Mathematical operations in FORTH

Week 2 : Fundamental Electronics
Robot Project Explanation
Read Chapter 2 in Starting FORTH
Project 1 Due
Assign Project 2
Stack Manipulations and FORTH Arithmetic

Week 3 : Project 2 Due
Read Chapter 3 in Starting FORTH
The FORTH Editor
Assign Project 3
Robot Construction

Week 4 : Project 3 Due
Read Chapter 4 in Starting FORTH
FORTH Decision Structures
Assign Project 4
Robot Construction (continued)

Week 5 : Project 4 Due
Read Chapter 5 in Starting FORTH
The FORTH Return Stack
Factoring Equations in FORTH
More Arithmetic Operations and Techniques
Assign Project 5
Robot Construction (conclusion)

Week 6 : Project 5 Due
Read Chapter 6 in Starting FORTH
FORTH Loop Structures
Assign Project 6
Robot Testing

Week 7 : Project 6 Due
Read Chapter 7 in Starting FORTH
Signed and Unsigned Numbers
Double and Single-length Numbers
ASCII Character Set
Double-length Operators
Mixed-length Operators
Number Formatting
Assign Project 7
Introduction to Robot Programming

Week 8 : Project 7 Due
Read Chapter 8 in Starting FORTH
Variables, Constants, and Arrays
Memory Usage
Hardware Interfacing Techniques with the Apple II
Assign Project 8
Midterm Review

Week 9 : Project 8 Due
Midterm
Read Chapter 9 in Starting FORTH
FORTH Vocabularies
Vectored Execution
The FORTH System
Assign Project 9
Primitive Robot Instructions Created in FORTH

Week 10: Project 9 Due
Read Chapter 10 in Starting FORTH
FORTH Input/Output Operations
String Manipulation
Assign Project 10
Introduction to Artificial Intelligence

Week 11: Project 10 Due
Read Chapter 11 in Starting FORTH
The FORTH Compiler
FORTH Defining Words
Assign Project 11
Introduction to Maze Solution Strategies
Searches

Week 12: Project 11 Due
Read Chapter 12 in Starting FORTH
Current Trends in Artificial Intelligence
Assign Artificial Intelligence Problem 1
Expert Systems

Week 13: Current Trends in Artificial Intelligence (continued)
Introduction to LISP
Eliza
Natural Language Processing

Week 14: Artificial Intelligence Problem 1 Due
Current Trends in Artificial Intelligence (continued)
Perception and Pattern Recognition
Information Storage and Retrieval
Animals
Assign Artificial Intelligence Problem 2

Week 15: Current Trends in Artificial Intelligence (continued)
Control of Robots
Introduction to Heath's Educational Robot (HERO 1)
Automatic Programming
Problem Solving
Game Playing
Sargon

Week 16: Artificial Intelligence Problem 2 Due
Current Trends in Artificial Intelligence (conclusion)
Summary
Final Review

Week 17: Final Examination (Post-test)

APPENDIX F

PROGRAMMING ASSIGNMENTS

AND

SOLUTIONS

FORTH PROJECT 1

Write the Forth instructions necessary to perform each of the following tasks.

1. Define a word called GIFT which, when executed, will type out the name of some gift. For example, you might try:

```
: GIFT  ." BOOKENDS " ;
```

Now define a word called GIVER which will print out a person's first name. Finally, define a word called THANKS which includes the new Forth words GIFT and GIVER, and prints out a message something like this:

```
DEAR STEPHANIE,
    THANKS FOR THE BOOKENDS. OK
```

2. After entering the definitions requested in the first task, enter a new definition for GIVER to print someone else's name, then execute THANKS again. Explain why THANKS still prints out the first giver's name.

3. Define a word called TEN-LESS which takes a number on the stack, subtracts ten, and returns the answer on the stack.

```
SCR # 8
  0 ( ASSIGNMENT #1 SOLUTION )
  1 : GIFT
  2     ." BOOKENDS" ;
  3 : GIVER
  4     ." GENE" ;
  5 : THANKS
  6     CR ." DEAR " GIVER CR
  7     4 SPACES ." THANKS FOR THE "
  8     GIFT ." . " ;
  9
 10 ( THANKS HAS BEEN COMPILED USING )
 11 ( THE DEFINITION SHOWN ON THIS )
 12 ( SCREEN.  CREATING A NEW GIVER )
 13 ( WILL NOT ALTER THE DEFINITION )
 14 ( OF THANKS. )
 15 -->

SCR # 9
  0 ( ASSIGNMENT #1 CONTINUED )
  1 : TEN-LESS
  2     10 - ;
  3 ;S
  4
  5
  6
  7
  8
  9
 10
 11
 12
 13
 14
 15
```

FORTH PROJECT 2

Write Forth instructions that will accomplish each of the following tasks.

1. Write a definition that will reverse the order of the top four items on the stack, that is,
(1 2 3 4 -- 4 3 2 1)

2. Write a definition called 3DUP which will duplicate the top three numbers on the stack; for example,
(1 2 3 -- 1 2 3 1 2 3)

3. Write definitions for the following infix equations, given the stack effects shown:

a^2 + ab + c	(c a b -- result)
a-b --- a+b	(a b -- result)
x(7x + 5)	(x -- result)
9a^2 - ba	(a b -- result)

```
SCR # 10
  0 ( ASSIGNMENT #2 SOLUTION )
  1 ( 1 2 3 4 -- 4 3 2 1 )
  2 : REVERSE SWAP 2SWAP SWAP ;
  3
  4 ( 1 2 3 -- 1 2 3 1 2 3 )
  5 : 3DUP DUP 2OVER ROT ;
  6
  7 ( C A B -- RESULT )
  8 : EQ1 OVER DUP * SWAP ROT * + + ;
  9
 10 ( A B -- RESULT )
 11 : EQ2 2DUP - SWAP ROT SWAP + / ;
 12
 13 ( X -- RESULT )
 14 : EQ3 DUP 7 * 5 + * ;
 15 -->

SCR # 11
  0 ( ASSIGNMENT #2 CONTINUED )
  1 ( A B -- RESULT )
  2 : EQ4 OVER DUP * 9 * SWAP ROT * - ;
  3 ;S
  4
  5
  6
  7
  8
  9
 10
 11
 12
 13
 14
 15
```

FORTH PROJECT 3

Write Forth definitions using the Forth Editor that solve the following problems. Make sure your definitions are stored on the disk. Turn in a hard copy of your definitions.

1. You're the inventory programmer at Paula's Egg Ranch. Define a word called EGG-CARTONS which expects on the stack the total number of eggs laid by the chickens today and prints out the number of cartons that can be filled with a dozen each, as well as the number of left-over eggs.

2. Write a definition which flips three items on the stack, leaving the middle number in the middle; that is,
   a b c becomes c b a

3. Write a definition that does what OVER does, without using OVER.

4. Write a definition called <ROT, which rotates the top three stack items in the opposite direction from ROT; that is,
   a b c becomes c a b

```
SCR # 12
  0 ( ASSIGNMENT #3 SOLUTION )
  1
  2
  3 ( GENE RECK )
  4 ( ROBOTICS - CAP2930 )
  5 ( WEDNESDAY NIGHTS )
  6 ( 7:00 TO 9:50 )
  7 ( GENE RECK )
  8 ( PROJECT #3 )
  9
 10 -->
 11
 12
 13
 14
 15

SCR # 13
  0 ( ASSIGNMENT #3 CONTINUED )
  1
  2 ( N1 -- N2 N3 )
  3 : EGG-CARTONS
  4      12 /MOD CR ." YOU WILL NEED " .
  5      ." EGG CARTONS," CR ." AND HAVE "
  6      . ." EGGS LEFT OVER." CR ;
  7
  8 ( A B C -- C B A )
  9 : FLIP
 10      SWAP ROT ;
 11
 12 ( N1 N2 -- N1 N2 N1 )
 13 : OVER-DOES
 14      SWAP DUP ROT SWAP ;
 15 -->

SCR # 14
  0 ( ASSIGNMENT #3 CONTINUED )
  1
  2 ( A B C -- C A B )
  3 : <ROT
  4      SWAP ROT SWAP ;
  5 ;S
  6
  7
  8
  9
 10
 11
 12
 13
 14
 15
```

FORTH PROJECT 4

Write Forth definitions using the Forth Editor that accomplish each of the following tasks. Turn in hard copies of your definitions.

1. Define a word called CARD which, given a person's age on the stack, prints out either of these two messages (depending on whether the person is over 21 or not):

```
ALCOHOLIC BEVERAGES PERMITTED  or
UNDER AGE
```

A sample run would go something like this:

```
21 CARD
ALCOHOLIC BEVERAGES PERMITTED OK

18 CARD
UNDER AGE OK
```

2. Define a word called SIGN-TEST that will test a number on the stack and print out one of three messages:

```
POSITIVE  or
ZERO  or
NEGATIVE
```

3. Using nested tests and IF...ELSE...THEN statements, write a definition called SPELLER which will spell out a number that it finds on the stack, from -4 to 4. If the number is outside this range, it will print the message "OUT OF RANGE". For example:

```
2 SPELLER
TWO  OK

-4 SPELLER
NEGATIVE FOUR  OK

7 SPELLER
OUT OF RANGE  OK
```

Make it as short as possible. (Hint: the Forth word ABS gives the absolute value of a number on the stack.)

```
SCR # 33
  0 ( ASSIGNMENT #4 SOLUTION )
  1
  2 ( N1 -- )
  3 : CARD
  4     CR 20 > IF ." ALCOHOLIC BEVERAGES PERMITTED "
  5               ELSE ." UNDER AGE "
  6               THEN ;
  7
  8 ( N1 -- )
  9 : SIGN-TEST
 10     CR DUP 0= IF ." ZERO" DROP
 11                ELSE 0 > IF ." POSITIVE"
 12                          ELSE ." NEGATIVE"
 13                          THEN
 14                THEN CR ;
 15 -->

SCR # 34
  0 ( ASSIGNMENT #4 CONTINUED )
  1
  2 ( N1 -- )
  3
  4 : SPELLER
  5     CR DUP ABS 4 > IF ." OUT OF RANGE"
  6                    ELSE DUP 0< IF ." NEGATIVE "
  7                                THEN
  8                       ABS DUP 4 = IF ." FOUR"
  9                                  ELSE DUP 3 = IF ." THREE"
 10                                   ELSE DUP 2 = IF ." TWO"
 11                                     ELSE DUP 1 = IF ." ONE"
 12                                       ELSE ." ZERO"
 13                                       THEN
 14                                     THEN
 15 -->

SCR # 35
  0 ( ASSIGNMENT #4 CONTINUED )
  1                                        THEN
  2                                      THEN
  3
  4                       THEN
  5                       DROP CR ;
  6 ;S
  7
  8
  9
 10
 11
 12
 13
 14
 15
```

FORTH PROJECT 5

Write Forth definitions that accomplish each of the following tasks. Turn in hard copies of each of these definitions.

1. Given four numbers on the stack, write an expression that prints the largest of the values.
   (a b c d -- max)

2. Given the following formulas:

   degrees C = (degrees F - 32) / 1.8

   degrees F = (degrees C * 1.8) + 32

   degrees K = degrees C + 273

   Define words with the following names that perform the appropriate conversions.
   F>C F>K C>F C>K K>F K>C

   For example,

```
32 F>C
0  OK

0 C>F
32  OK
```

```
SCR # 20
  0 ( ASSIGNMENT #5 SOLUTION )
  1
  2 ( A B C D -- MAX )
  3 : BIGGEST
  4     MAX >R MAX R> MAX DUP . ;
  5
  6 ( N1 -- N2 )
  7 : (F>C)
  8     32 - 10 18 */ ;
  9 : (F>K)
 10     (F>C) 273 + ;
 11 : (C>F)
 12     18 10 */ 32 + ;
 13 : (C>K)
 14     273 + ;
 15 -->

SCR # 21
  0 ( ASSIGNMENT #5 CONTINUED )
  1
  2 : (K>C)
  3     273 - ;
  4 : (K>F)
  5     (K>C) (C>F) ;
  6
  7 : F>C CR (F>C) . ;
  8 : F>K CR (F>K) . ;
  9 : C>F CR (C>F) . ;
 10 : C>K CR (C>K) . ;
 11 : K>F CR (K>F) . ;
 12 : K>C CR (K>C) . ;
 13
 14
 15 ;S
```

FORTH PROJECT 6

Write Forth definitions that perform the specified tasks. Turn in a hard copy of your definitions.

1. Create a word called STARS which will print out n stars on the same line, given n on the stack:

```
10 STARS ********** OK
```

2. Next define BOX which prints out a rectangle of stars, given the width and height (number of lines), using the stack order (width height --).

```
10 3 BOX
**********
**********
********** OK
```

3. Define a word called POWER that will compute exponential values, like this:

```
7 2 POWER . 49 OK          (seven squared)

2 4 POWER . 16 OK          (two to the fourth power)
```

For simplicity, assume positive exponents only (but make sure POWER works correctly when the exponent is one - the result should be the number itself).

```
SCR # 24
  0 ( ASSIGNMENT #6 SOLUTION )
  1 ( N -- )
  2 : STARS
  3     0 DO ." *" LOOP ."   " ;
  4
  5 ( WIDTH HEIGHT -- )
  6 : BOX
  7     0 DO DUP CR STARS LOOP DROP ;
  8
  9 ( BASE EXPONENT -- POWER )
 10 : POWER
 11     DUP NOT IF 2DROP 1
 12              ELSE DUP 1 =
 13                IF DROP
 14                ELSE OVER SWAP
 15                  1 DO OVER * LOOP -->

SCR # 25
  0 ( ASSIGNMENT #6 CONTINUED )
  1                   SWAP DROP
  2                 ENDIF
  3               ENDIF ;
  4 ;S
  5
  6
  7
  8
  9
 10
 11
 12
 13
 14
 15
```

FORTH PROJECT 7

Write Forth definitions that accomplish the assigned tasks. Turn in hard copies of your definitions.

1. Mary Beth Mnemonic couldn't remember the upper limit for a signed single-length number, and she had no book to refer to, only a Forth terminal. So she wrote a definition called N-MAX, using a BEGIN...UNTIL loop. When she executed it, she got "32767 OK". What was her definition?

2. Write a word which prints the numbers 0 through 16 (decimal) in decimal, hexadecimal, and binary form in three columns. For example,

```
DECIMAL   0   HEX   0   BINARY       0
DECIMAL   1   HEX   1   BINARY       1
   .      .     .   .      .         .
DECIMAL  16   HEX  FF   BINARY   10000
```

3. Write a definition that "rings" your terminal's bell three times. Make sure that there is enough of a delay between the bells so that they are distinguishable. Each time the bell rings, the word "BEEP" should appear on the terminal screen. Do not use the predefined word BELL in your definition.

```
SCR # 26
  0 ( ASSIGNMENT #7 SOLUTION )
  1
  2 ( -- N )
  3 : N-MAX
  4     0 BEGIN 1 + DUP 0 < UNTIL 1 -
  5     . SPACE ;
  6
  7 ( -- )
  8 : CONVERTER
  9     17 0 DO CR DECIMAL
 10     ." DECIMAL   " I 2 .R 3 SPACES
 11     ." HEX     " HEX I 2 .R 3 SPACES
 12     ." BINARY    " BINARY I 5 .R
 13     LOOP DECIMAL CR ;
 14
 15 -->

SCR # 27
  0 ( ASSIGNMENT #7 CONTINUED )
  1
  2 (  --  )
  3 : DELAY 5000 0 DO LOOP ;
  4
  5 : RING
  6     3 0 DO CR 7 EMIT ." BEEP"
  7             DELAY LOOP CR ;
  8 ;S
  9
 10
 11
 12
 13
 14
 15
```

FORTH PROJECT 8

Write Forth definitions that accomplish each of the following assigned tasks. Turn in hard copies of your definitions.

1. Define a word called .BASE which prints the current value of the variable BASE in decimal. Test it by first changing BASE to some value other than ten.

```
DECIMAL .BASE 10 OK
HEX .BASE 16 OK
BINARY .BASE 2 OK
```

2. A histogram is a graphic representation of a series of values. Each value is shown by the height or length of a bar. In this exercise you will create an array of values and print a histogram which displays a line of "*"s for each value. First create an array with about ten cells. Initialize each element of the array with a value in the range zero to thirty. Then define a word GRAPH which will print a line for each value. On each line print the number of the cell followed by a number of "*"s equal to the contents of that cell. For example, it the array has four cells and contains the values 1, 2, 3, and 4, then GRAPH would produce:

```
1 *
2 **
3 ***
4 ****
```

```
SCR # 28
  0 ( ASSIGNMENT #8 SOLUTION )
  1 : .BASE
  2     BASE @ DECIMAL DUP . BASE ! ;
  3
  4 0 VARIABLE STORAGE 18 ALLOT
  5 : INITIALIZE
  6     10 0 DO 30 RND STORAGE I 2 * + !
  7          LOOP ;
  8 : GRAPH  CR CR
  9     10 0 DO CR 2 SPACES I 1 + 2 .R
 10              STORAGE I 2 * + @
 11              DUP IF SPACE 0 DO 42 EMIT
 12                           LOOP
 13                   THEN
 14          LOOP CR CR ;
 15 ;S
```

FORTH PROJECT 9

Answer each of the following questions based on information you are able to obtain from Forth about itself.

1. What is the NFA of BELL?

2. What is the CFA of SOUND?

3. What is the PFA of DUMP?

4. Explain the setup of a typical Forth definition.

```
SCR # 36
 0 ( ASSIGNMENT #9 SOLUTION )
 1
 2 CR ." THE NFA OF BELL IS "
 3       ' BELL NFA .
 4       ( -32163 OR 33373 )
 5
 6 CR ." THE CFA OF SOUND IS "
 7       ' SOUND CFA .
 8       ( -32193 OR 33343 )
 9
10 CR ." THE PFA OF DUMP IS "
11       ' DUMP .
12       ( -32126 OR 33410 )
13 -->
14
15

SCR # 37
 0 ( ASSIGNMENT #9 CONTINUED )
 1
 2 ( ------------------------------------------------------ )
 3 ( ! NAME FLD ! LINK FLD ! CODE FLD ! PARAMETER FIELD ! )
 4 ( ------------------------------------------------------ )
 5 ;S
 6
 7
 8
 9
10
11
12
13
14
15
```

FORTH PROJECT 10

Write Forth definitions that accomplish each of the following tasks. Turn in hard copies of your definitions.

1. Enter some famous quotations into an available block using the Forth Editor, say block 99. Now define a word called CHANGE which takes two ASCII values and changes all occurrences within block 99 of the first character into the second character. For example,

```
65 69 CHANGE
```

will change all the "A"s into "E"s.

2. Define a word called FORTUNE which will print a prediction at your terminal, such as "YOU WILL RECEIVE GOOD NEWS IN THE MAIL". The prediction should be chosen at random from a list of sixteen or fewer predictions. Each predicition should be sixty-four characters, or less, long.

```
SCR # 30
  0 ( ASSIGNMENT #10 SOLUTION )
  1
  2 ( N1 N2 -- )
  3 ( FIRST 4 LINES OF SCREEN #50 )
  4
  5 : CHANGE
  6      200 BLOCK ROT
  7      256 0 DO
  8            OVER I + C@ OVER =
  9            IF 0 2OVER I + C! DROP
 10            THEN
 11      LOOP
 12      2DROP DROP ;
 13 ;S
 14
 15

SCR # 31
  0 ( ASSIGNMENT #10 CONTINUED )
  1
  2 ( -- )
  3 ( PREDICTIONS FOUND ON SCREEN #51 )
  4
  5 : FORTUNE
  6      16 RND 4 /MOD 51 4 * + BLOCK
  7      SWAP 64 * + 64 CR TYPE ;
  8 ;S
  9
 10
 11
 12
 13
 14
 15
```

```
SCR # 50
  0 WHEN IN THE COURSE OF HUMAN EVENTS
  1 FOUR SCORE AND SEVEN YEARS AGO
  2 OUR FATHERS BROUGHT FORTH ONTO THIS CONTINENT
  3 A NEW NATION, CONCEIVED IN LIBERTY.
  4
  5
  6
  7
  8
  9
 10
 11
 12
 13
 14
 15

SCR # 51
  0 ALL WORK AND NO PLAY MAKES MIKE A DULL BOY
  1 YOU WILL RECEIVE GOOD NEWS IN THE MAIL
  2 YOU WILL MEET A TALL, DARK STRANGER
  3 GOOD THINGS COME TO HE WHO WAITS
  4 GOOD THINGS ARE COMING YOUR WAY
  5 BEWARE OF STRANGERS
  6 NOW IS THE BEST TIME OF YOUR LIFE
  7 CONSIDER THE FUTURE
  8 BEWARE THE IDES OF MARCH
  9 LOOK BEFORE YOU LEAP
 10 DON'T BE AFRAID TO TAKE RISKS
 11 IF AT FIRST YOU DON'T SUCCEED - QUIT
 12 YOU SHOULD HAVE STAYED IN BED!
 13 IS YOUR INSURANCE PAID UP?
 14 DON'T TAKE ANY WOODEN NICKELS
 15 THE BEST IS YET TO COME
```

FORTH PROJECT 11

Write a Forth definition for the following task. Turn in a hard copy of this definiition.

1. Define a defining word BASED. which will create number output words for specific bases. For example,

```
16 BASED. H.
```

would define H. to be a word which prints the top of the stack in Hex but does not permanently change BASE.

```
DECIMAL
17 DUP H. . 11 17 OK
```

```
SCR # 32
  0 ( ASSIGNMENT #11 SOLUTION )
  1
  2 ( N -- )
  3 : BASED <BUILDS , DOES> @
  4       BASE @ SWAP BASE ! SWAP .
  5       BASE ! ;
  6 ;S
  7
  8
  9
 10
 11
 12
 13
 14
 15
```

ARTIFICIAL INTELLIGENCE PROBLEM 1

Write an application in Forth that allows your robot to successfully navigate its way through a maze with the following configuration:

Turn in a hard copy of the definitions used in your solution.

```
SCR # 22
  0 ( ROBOT TEST INSTRUCTIONS )
  1
  2 49240 CONSTANT A0
  3 A0 2 + CONSTANT A1
  4 A1 2 + CONSTANT A2
  5 A2 2 + CONSTANT A3
  6
  7 : FORWARD
  8     0 A0 1 + C! 0 A1 C! 0 A2 1 + C!
  9     0 A3 C! ;
 10
 11 : BACKWARD
 12     0 A0 C! 0 A1 1 + C! 0 A2 C!
 13     0 A3 1 + C! ;
 14
 15 --->

SCR # 23
  0 ( ROBOT TEST INSTRUCTIONS CONT. )
  1
  2 : LEFT-TURN
  3     0 A0 C! 0 A1 1 + C! 0 A2 1 + C!
  4     0 A3 C! ;
  5
  6 : RIGHT-TURN
  7     0 A0 1 + C! 0 A1 C! 0 A2 C!
  8     0 A3 1 + C! ;
  9
 10 : STOP
 11     0 A0 C! 0 A1 C! 0 A2 C!
 12     0 A3 C! ;
 13
 14 : SENSE 49249 C@ ;
 15 ;S
```

```
SCR # 38
  0 ( A.I. ASSIGNMENT #1 SOLUTION )
  1
  2 : BACK-UP
  3     BACKWARD 5000 0 DO LOOP STOP ;
  4
  5 : GO-TO-WALL
  6     FORWARD BEGIN SENSE 127 < UNTIL STOP BACK-UP ;
  7
  8 : FACE-RIGHT
  9     RIGHT-TURN 5000 0 DO LOOP STOP ;
 10
 11 : FACE-LEFT
 12     LEFT-TURN 7500 0 DO LOOP STOP ;
 13 -->
 14
 15

SCR # 39
  0 ( A.I. ASSIGNMENT #1 CONTINUED )
  1
  2 : RUN-MAZE
  3     GO-TO-WALL
  4     FACE-LEFT
  5     GO-TO-WALL
  6     FACE-LEFT
  7     GO-TO-WALL
  8     FACE-RIGHT
  9     GO-TO-WALL
 10     FACE-RIGHT
 11     GO-TO-WALL ;
 12 ;S
 13
 14
 15
```

ARTIFICIAL INTELLIGENCE PROBLEM 2

Write a generalized maze solving application that will solve as many different types of mazes as possible. Be prepared to test your solution. Turn in a hard copy of the definitions used by your application. If for some reason you are unable to solve this problem, explain why not, what strategies you considered, what problems and limitations you encountered, and any other points that should be considered.

```
SCR # 40
  0 ( A.I. ASSIGNMENT #2 SOLUTION )
  1
  2 ( THERE WAS NO CORRECT SOLUTION TO THIS PROBLEM AS )
  3 ( PRESENTED DUE TO THE LIMITED SENSING CAPABILITIES )
  4 ( OF THE ROBOT.  SEVERAL MAZE SOLUTION STRATEGIES )
  5 ( COULD BE PROPOSED, BUT ALL WOULD FAIL BECAUSE )
  6 ( THE ROBOT COULD NOT DETECT WHEN IT HAD SUCCEEDED )
  7 ( IN SOLVING THE MAZE.  IN ANY MAZE TYPE OF PROBLEM )
  8 ( THE ROBOT MUST BE ABLE TO COMPARE ITS PRESENT )
  9 ( CONDITION WITH SOME DESIRED GOAL STATE.  PUT )
 10 ( SIMPLY, THE ROBOT COULD NOT TELL IF IT HAD )
 11 ( SOLVED THE MAZE. )
 12
 13 ( SOLUTIONS THAT COULD HAVE BEEN PROPOSED INCLUDE )
 14 ( THOSE THAT CONTINUALLY FOLLOW EITHER THE RIGHT )
 15 ( OR THE LEFT WALLS UNTIL AN ESCAPE ROUTE IS FOUND. )

SCR # 41
  0 ( A.I. ASSIGNMENT #2 CONTINUED )
  1
  2 ( THE CLASS ROBOT DOES NOT HAVE THE ACCURACY )
  3 ( NECESSARY FOR THIS TYPE OF SOLUTION STRATEGY. )
  4 ( MORE SENSORS COULD BE ADDED TO REMEDY THIS )
  5 ( SHORTCOMING.  THE CABLE ATTACHING THE ROBOT )
  6 ( TO THE COMPUTER ALSO LIMITS THE FREEDOM OF )
  7 ( THE ROBOT.  TWO MEANS OF ALLOWING THE ROBOT )
  8 ( TO DETECT THE END OF MAZE CONDITION WOULD BE )
  9 ( TO EITHER PROVIDE A WAY OUT OF THE MAZE AND )
 10 ( HAVE THE ROBOT CHECK FOR WALLS FOR A CERTAIN )
 11 ( TIME PERIOD OR ADD A SENSOR THAT COULD )
 12 DETECT AN END OF MAZE CONDITION. )
 13
 14 ( STUDENT ANSWERS SHOULD REFLECT THE COMPLEXITY )
 15 ( AND DIFFICULTY OF THE PROBLEM. )
```

APPENDIX G

SUPPORT MATERIALS

MIDTERM EXAMINATION

Name _____

Section 1 (25 points)

What are the major advantages and disadvantages to FORTH as a programming language? How does it compare to other programming languages?

Section 2 (25 points)

Explain the theory behind the operation of the educational robot built in class. Include relevant information about the interface and the microcomputer that are attached to the robot.

Section 3 (25 points)

Describe the historical development of robots. Be sure to include the types of robots currently in use and their role in our society.

Section 4 (25 points)

Write a FORTH definition that accepts two numbers from the stack and counts from the second to the first. For example,

```
5 1 COUNT
1
2
3
4
5 OK
```

EXTRA CREDIT (5 points)

Which country currently leads the world in use of robots?

ANSWER KEY FOR MIDTERM EXAMINATION

1. Advantages of FORTH include the following:

- Interactive
- Encourages structured programming
- Flexible
- Good control structures
- Short program development time
- Increased programmer productivity
- Highly efficient
- Fast
- Compact
- Transportable
- Extensible
- Absolute control over hardware

Disadvantages of FORTH include the following:

- Harder to learn than most other languages
- Limited error checking
- Postfix Arithmetic
- Integer Arithmetic Only
- Hard to read FORTH programs
- Hard to debug
- Hogs the system
- No real standard

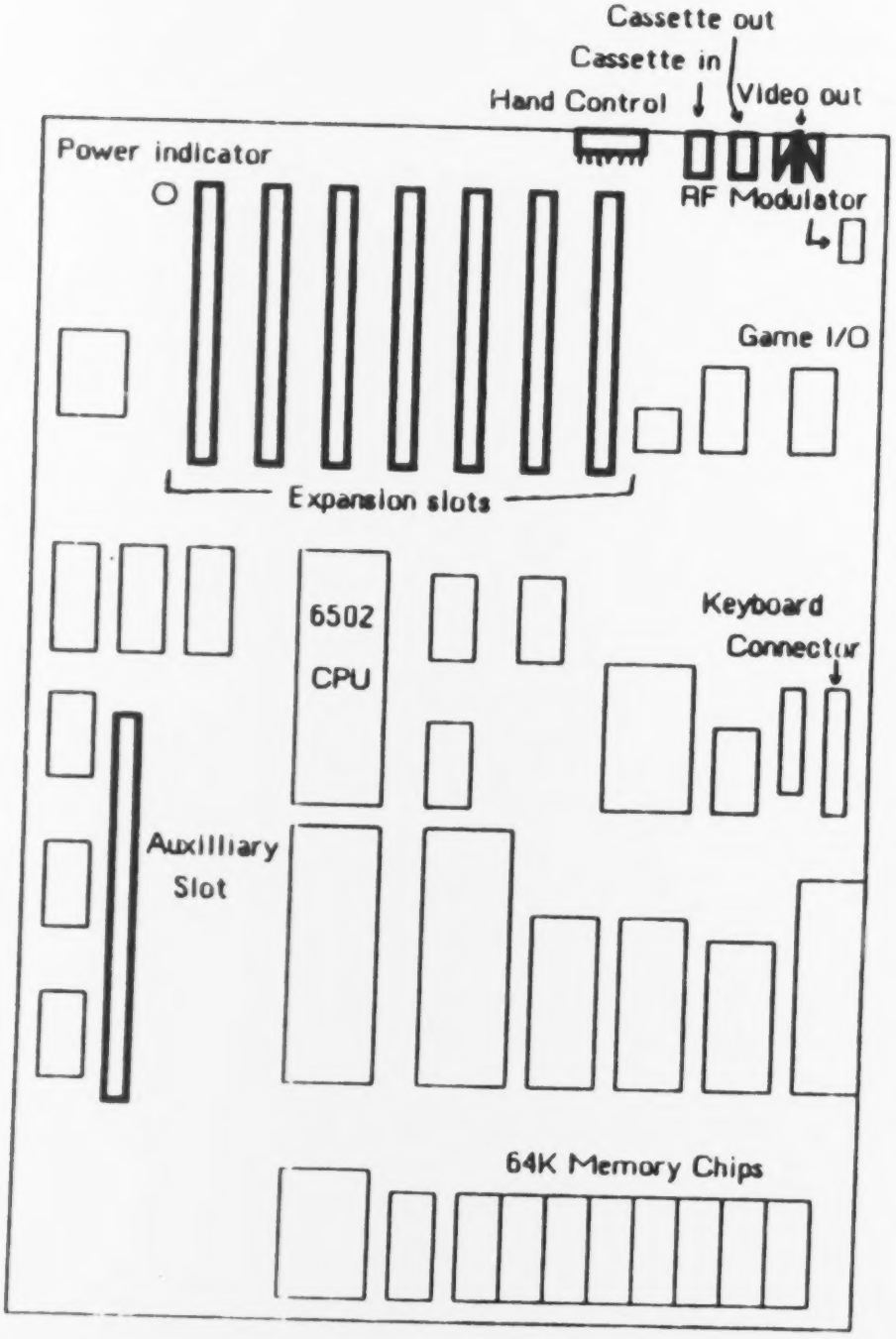
2. The robot constructed in class used two inexpensive battery powered cars as actuators that were controlled by an Apple II personal computer through four relays. These motors allowed the robot to move forward, backward, left, right, and stop under stored program control. A switch mounted on the front of the robot acts as a sensing mechanism that allows detection of contact between the robot and an object in front of it. The robot uses a common DB-25 connector as part of the interface cable that connects the robot to the computer. A 74LS367 integrated circuit is used to boost the power coming from the computer to a level where it is sufficient to activate the relays. The robot moves by activating the relays connected to the car motors. The cars were mounted in opposite directions to provide additional stability.

3. The historical trend has been from mechanization to automation to robots. Robots were derived from hard automation. Robots have been primarily found in industrial applications thus far. The word robot originated in a play written during the 1920's by Karel Capek titled R.U.R., for Rosom's Universal Robots. The word robot means "to drudge". In the 1950's Dr. W. Walter Grey developed small tortoise shaped robots that were able to move around a room, avoid obstacles, and recharge themselves. Also during the 1950's, George Devol and Joseph Engelberger joined forces to found Unimation, the first and largest company to produce robots for industry. The greatest advance in the evolution of the robot was the adaptation of microprocessor-based computer control systems. Currently Japan leads the world in the production and use of robots.

4. : COUNT
SWAP 1 + SWAP DO CR I . LOOP ;

Extra Credit
Japan

Apple IIe Motherboard



Electronic Review Sheet

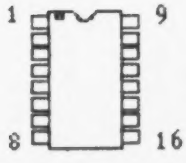
CAP 2930 - Robotics
Gene Reck
Wednesday Nights 7:00 to 9:50 P.M.

Resistor
An electrical component used in electronic circuits to resist the flow of electricity.

Diode
An electrical component used in electronic circuits to demodulate alternating current into direct current.

Ground
That part of an electrical circuit whose potential is regarded as zero and to which an electrical current may be conducted.

Relay
An electromechanical device in which contacts are opened and/or closed by variations in the conditions of an electric circuit, thereby affecting the operation of other devices in the same or other circuits.



Chips
Chips are integrated circuits which combine a number of electronic components into a single unit. Each chip may be identified by a unique number printed on the top. Chips must be oriented in the proper direction to function correctly.

Microprocessor
The chip that performs the arithmetic and logical operations within a microcomputer. The Apple II+ and Apple IIe computers use a 6502 microprocessor.

CMOS
Complementary Metal Oxide Semiconductor
A type of computer chip known for its low heat production and low power consumption. Especially useful where portable computing functions are required.

RAM
Random Access Memory
A type of computer memory that goes blank when the computer's power is turned off. Read/Write Memory.

ROM
Read Only Memory
A type of computer memory that is permanent and cannot be changed.

ASCII
American Standard Code for Information Interchange
The collating sequence used by most microcomputers to represent characters.

Bit
A binary digit, either a 0 or a 1. May be thought of as an on or an off.

Byte
A number composed of eight binary digits, or bits. The fundamental unit of computer memory storage.

K (Kilobyte)
A term referring to exactly 1024 bytes. Often rounded off to one thousand For example, 16 K refers to exactly 16,384 bytes (16 X 1024), or approximately 16,000 bytes.

ADVANTAGES OF FORTH

1. Interactive
2. Encourages structured programming
3. Flexible
4. Good control structures
5. Short program development time
6. Increased programmer productivity
7. Highly efficient
8. Fast
9. Compact
10. Transportable
11. Extensible
12. Absolute control over the hardware

DISADVANTAGES OF FORTH

1. Harder to learn than most other languages
2. Limited error checking
3. Reverse Polish Notation
4. Integer Arithmetic Only
5. Hard to read FORTH programs
6. Hard to Debug
7. Hogs the System
8. No real standard

Saving Extended Versions of FORTH

The following commands will allow you to save new versions of FORTH to disk with the definitions you have added made a permanent part of the system.

After defining your new words, type:

```
FORTH DEFINITIONS DECIMAL <RETURN>

LATEST 12 +ORIGIN ! <RETURN>
                        /* Alter pointer to topmost
                                word in dictionary  */

HERE 28 +ORIGIN ! <RETURN>
                        /* Alter FENCE pointer      */

HERE 30 +ORIGIN ! <RETURN>
                        /* Alter value of Dictionary
                                pointer             */
```

To save this version to disk, you will need to know the starting address and the number of bytes to save. The starting address is Hexadecimal $6000. The length in bytes may be determined by the following command:

```
HEX HERE 6000 - . <RETURN>
```

Now type:

```
MON <RETURN>            /* Get to Apple Monitor
                                program             */

BSAVE NEWSYSTEMNAME,A$6000,L$length from above <RETURN>
```

Your new FORTH system will now be saved to whatever initialized disk was in the current drive with the name you provided as the new system name. To activate your new version, type:

```
BRUN NEWSYSTEMNAME <RETURN>
```

at either the Applesoft prompt] or the monitor prompt *.

Robotics

1. Most robots today are found in industrial applications.

2. The word "Robot" comes originally from a Czech word meaning "to drudge". First used in a play written by Karel Capek titled "R.U.R." (Rosom's Universal Robots). In this play robots were depicted as mechanical men in a slave-like position who overthrew their human masters.

3. In the 1950's, Dr. W. Walter Grey developed small robots shaped like tortoises that were able to move around a room, avoid obstacles, and recharge their batteries when they were low.

4. George Devol and Joseph Engleberger joined forces to found Unimation, the first and largest company to produce robots for industry. These robots appeared much like most modern industrial robots found today, a base with a moveable arm and hand.

5. Modern industrial robots evolved from "hard automation", that is, equipment designed to accomplish repetitive production-line tasks at a fast pace. Hard automation is usually very specific. Any change in production is likely to require a mechanical modification.

6. Servo controls are feedback devices that generate position signals. Servo control systems compare the input signal to the output signal, and when these two signals coincide, the robot ceases movement.

7. The greatest advance in the evolution of the robot was the adaptation of microprocessor-based computer control. These low cost controllers brought about the rapid development of special purpose robots.

Robotic Terms

Axes - The number of intricate motions a robot can perform is determined by the number of axes the robot has. Also known as Degrees of Freedom. The complexity of a task that a robot can perform is determined by the number of axes the robot has. Most industrial applications require only three to five degrees of freedom to perform a specific task.

Manipulator - A system of mechanical linkages and joints that can be moved in various directions to perform the work of the robot. In some instances the manipulator is also called the robot arm.

Actuators - The drive mechanisms that position the manipulator to a predetermined point. Actuators may consist of electric motors, pneumatic or hydraulic rotary motors, or pneumatic or hydraulic cylinders.

Gripper - A hand-like device sometimes referred to as the "end-effector". Often used to hold tools or materials.

Controller - The brains and nervous system of the robot. A controller can be any programmable device from a simple mechanical rotary drum switch to a complex electronic computer.

Power Supply - Provides and regulates the energy for the robot's actuators.

Cylindrical Coordinate Robot - Essentially, a horizontal manipulator mounted on a vertical column. The vertical column, in turn, is mounted on a rotating base. The manipulator is free to extend and retract, and is also able to move up and down on the vertical column. It can also rotate on the base. Thus, the work envelope (working area) of the robot forms a portion of a cylinder.

Spherical Coordinate Robot - Similar to the turret of a tank. The manipulator extends and retracts, pivots in the vertical plane, and rotates in the horizontal plane about the base. The work envelope forms a small portion of a sphere.

Jointed-Spherical Coordinate Robot - A base plus an upper arm and forearm which move in the vertical plane through the base. Manipulator often includes a "shoulder" and an "elbow". The work envelope forms a major portion of a sphere.

Non-Servo Robots - Robots that are restricted to three or four degrees of freedom. Mechanical stops (end stops) are used on each axis to limit the amount of travel. Thus, there are usually only two positions that each axis may be instructed to assume: up/down, left/right, extend/retract.

Servo-Controlled Robots - Robots whose axes of motion are controlled to move and to stop anywhere within their limits of travel, rather than only at the extremes. There are two classes of servo-controlled robots: point-to-point and continuous path.

Point-To-Point Servo Control - Allows the robot to move between precise, programmed, three-dimensional points within its work envelope. Point-to-point operation is always a little jerky. For this reason, point-to-point control is used where only the final position is important and the path and velocity between programmed points are not major considerations.

Continuous Path Servo Control - Allows the robot to smoothly follow a particular path rather than accelerate sharply between end stops. Ideally suited for spray painting applications. A major drawback is the large amount of controller memory required.

Low Technology Robots

(Pick and Place)
(limited Sequence)
Simplest type of robot.
Accounts for about 35% of all U.S. industrial robots.
Non-servo operation - Use End Stops.
Few degrees of freedom (usually 2 to 4)
Low payload capacity
High degree of accuracy
Greater speed
Most often pneumatically actuated (air driven)
Cannot learn - mechanical or electronic controllers

Medium Technology Robots

(Used in machine load/unload operations or material
transfer operations)
(Point-to-point operation)
Some use end stops, others may use servos.
Greater degrees of freedom (up to 6)
Greater payload capacity
Slower than low technology robots
Not as accurate as low technology robots
Usually actuated by hydraulic or electro-hydraulic means
Capability to Learn - mechanical or electronic controllers

High Technology Robots

sophisticated, programmable, microprocessor-controlled
Greatest degrees of freedom (6 and up)
Greatest payload capacity
Slower than lower technology robots
Accuracy and repeatability enhanced by optional sensory
mechanisms
May be actuated by pneumatic, hydraulic, or electrical means
Highly developed controller - may include sensory devices
High level programming languages

Robot Review Sheet

1. The real robots of today are the _____ robots.

2. Industrial robots have emerged from _____.

3. Hard automation devices are (seldom/frequently) altered to change their operation.

4. When a change to a hard automation device is required, it is usually a _____ adjustment.

5. Servo controls are _____ devices that generate position signals.

6. Servo control systems _____ the input signal to the output signal.

7. The greatest advancement in the evolution of the robot was the adaptation of _____ control.

8. Because of their low cost, _____ controllers brought about the rapid development of special purpose robots.

9. The number of _____ a robot has determines the complexity of a task the robot can perform.

10. The _____ is a system of mechanical linkages and joints that performs the robot's work.

11. The _____ drive the robot to a predetermined point.

12. The _____ is used to interface the robot with the outside world.

13. A _____ _____ robot has a work envelope that is a portion of a cylinder.

14. A jointed-spherical coordinate robot has an _____ joint plus a _____ joint to permit greater manipulator flexibility.

15. On non-servo robots, _____ are used to limit the amount of travel in each axis.

16. The two classes of servo-controlled robots are _____ and _____ _____.

17. In point-to-point control of a robot, the _____ can be set or taught anywhere in a given axis.

18. When a robot is being controlled by continuous path servo-control, a large amount of controller _____ is required.

19. Virtually all low-technology robots fall into the _____ robot class.

20. In low-technology robots, there are usually only _____ positions for each axis to assume.

21. The maximum distance of travel along each axis of a low-technology robot is usually (less/greater) than that of the more sophisticated robot.

22. The payload a robot can effectively carry is measured at the _____.

23. Most low-technology robots have a (faster/slower) cycle time than the more sophisticated robots.

24. When we speak of a robot's accuracy, we refer to _____ as the robot's ability to return to a given point when recycled.

25. Most pick-and-place robots are _____ actuated due to their light payloads.

26. Hydraulic actuation is generally (more/less) expensive than pneumatic actuation.

27. Because the low-technology robot has a very limited amount of memory available, usually only _____ and _____ are used in its program.

28. The _____ control system uses a series of small valves positioned on a sequencing module.

29. The _____ is the robot's most essential component.

30. Most medium-technology robots have (more/less) degrees of freedom than simple robots.

31. The amount of time a machine requires to perform a specific operation directly affects the _____ time of the robot serving that machine.

32. Two factors that greatly affect a robot's accuracy are _____ and _____.

33. Most medium-technology robots are _____ actuated.

34. The (low/medium) technology robot does not have the capability to learn.

35. The "rotating drum" controller (can/cannot) be used to store a specific program.

36. The _____ robots are considered to be the elite members of the robot family.

37. Since high-technology robots can work at the same _____ as humans, they fit easily into existing production operations.

38. Most high-technology robots use a _____ for control.

39. The _____ robot has the capability of carrying the greatest payload.

ANSWER KEY FOR ROBOT REVIEW SHEET

1. industrial
2. hard automation
3. seldom
4. mechanical
5. feedback
6. compare
7. computer
8. microprocessor
9. axes or degrees of freedom
10. manipulator
11. actuators
12. controller
13. cylindrical coordinate
14. elbow, shoulder
15. mechanical end stops
16. point to point, continuous path
17. end stops
18. memory
19. non-servo
20. two
21. less
22. gripper
23. faster

24. repeatability
25. pneumatically
26. more
27. sequence, time
28. air-logic
29. controller
30. more
31. cycle
32. speed, payload
33. hydraulic
34. low
35. can
36. high technology
37. speed
38. computer or microprocessor
39. high technology

Isaac Asimov's Three Laws of Robotics
from
I, Robot

1. A robot may not injure a human being, or, through inaction, allow a human being to come to harm.

2. A robot must obey orders given it by human beings except where such orders would conflict with the First Law.

3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

KAREL THE ROBOT MAZE SOLUTION STRATEGY

```
BEGINNING-OF-PROGRAM

        DEFINE-NEW-INSTRUCTION turnright AS
        BEGIN
                ITERATE 3 TIMES
                        turnleft
        END;

        DEFINE-NEW-INSTRUCTION corner-right AS
        BEGIN
                turnright;
                move
        END;

        DEFINE-NEW-INSTRUCTION left-or-straight AS
        BEGIN
                IF front-is-blocked
                        THEN turnleft
                        ELSE move
        END;

        DEFINE-NEW-INSTRUCTION search-for-beeper AS
        BEGIN
                IF right-is-clear
                        THEN corner-right
                        ELSE left-or-straight
        END;

        BEGINNING-OF-EXECUTION
                WHILE not-next-to-a-beeper
                        search-for-beeper;
                turnoff
        END-OF-EXECUTION
END-OF-PROGRAM
```

HERO 1 SPECIFICATIONS

SENSES

Sound: Frequency range of 200 to 5000 Hz.

Light: Sensative to visible spectrum.

Ultrasonic Ranging: Pulsed ultrasonic sonar system.

Motion: Continuous wave ultrasonic sonar system.

SPEECH

Synthesized Phoneme-based system. 64 phonemes in any combination.
Optional.

TIME

Built-in four year calendar clock.

ELECTRONIC COMPONENTS

Microprocessor - Motorola 6808 8-bit.

Memory - 8K ROM
4K RAM expandable to 56K

PHYSICAL COMPONENTS

HEAD - Contains most sensors. Rotates up to 350 degrees. Hex programming keypad.

ARM - Maximum payload is 1 pound. Contains five motors. Optional.

TORSO - Mounting surfaces for circuit boards. Housing for wiring harness.

BASE - Contains the mechanical components of the main drive and steering mechanisms.

BATTERIES - Four 4-ampere-hour rechargeable gel cells, connected in two electrically independent 12-volt systems (logic system and drive system).

CHARGER - Used to recharge the onboard batteries.

GENERAL

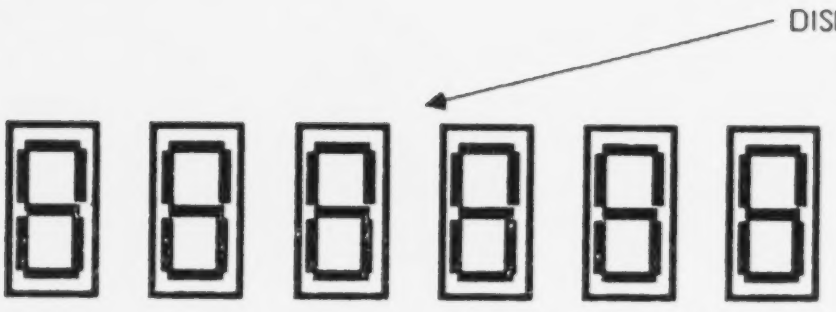
Operating Temperature - 32 to 104 degrees F.

Weight - 39 lbs. with accessories.

Dimensions - 20" high by 18" diameter (max.).

Minimum Turning Radius - 12".

HERO 1 KEYBOARD LAYOUT



	DO D	EXAM E	FWD F	USER 3
REPEAT	AUTO A	BACK B	CHAN C	USER 2
LEARN	RTI 7	SS 8	BR 9	USER 1
MANUAL	INDEX 4	CC 5	SP 6	
PROGRAM	ACCA 1	ACCB 2	PC 3	
	0	RESET		

KEYBOARD

BASIC LISTING OF ELIZA PROGRAM

```
100 REM ELIZA
110 PQ=VAL(RIGHT$(TIME$,2))
120 RANDOMIZE PQ
130 GOSUB 1250:REM INITIALIZE
140 PRINT:PRINT "------------------------------------------------------------
    ------------------":PRINT
150 LINE INPUT ">";X$
160 IF X$="" THEN END:REM QUIT BY JUST PRESSING 'RETURN'
170 PRINT
180 IF X$=Z$ THEN PRINT "PLEASE DON'T REPEAT YOURSELF":GOTO 140
190 Z$=X$
200 IF LEFT$(X$,7)="GOODBYE" THEN PRINT "OK, SEE YOU AGAIN SOMETIME":END
210 REM *** LOOK FOR TRIGGER PHRASES AT START OF INPUT ***
220 L=0
230 L=L+1
240 LN=LEN(C$(L))
250 IF LEFT$(X$,LN)=C$(L) THEN 470:REM TRIGGER FOUND
260 IF L<K THEN 230
270 REM *** PROGRAM GETS HERE IF NO TRIGGER PHRASE FOUND AT START OF X$ ***
280 REM *** NOW LOOK FOR TRIGGER WORDS WITHIN INPUT ***
290 X$=" "+X$+" "
300 M=LEN(X$)
310 L=0
320 L=L+1
330 IF L=M-1 THEN 910:REM NO TRIGGER FOUND
340 IF MID$(X$,L,1)=" " THEN 360
350 GOTO 320
360 X=L+1
370 Y=0
380 Y=Y+1
390 IF MID$(X$,(X+Y),1)=" " THEN Q$=MID$(X$,X,Y):GOTO 410
400 GOTO 380
410 N=0
420 N=N+1
430 IF Q$=C$(N) THEN 820:REM TRIGGER WORD FOUND
440 IF N<K THEN 420
450 GOTO 320
460 REM **********************
470 REM TRIGGER PHRASE FOUND AT START OF INPUT
480 T=INT(RND(1)*3)+1
490 IF T=1 THEN G$=D$(L)
500 IF T=2 THEN G$=E$(L)
510 IF T=3 THEN G$=F$(L)
520 REM *** CHECKS TO SEE IF ENDS IN ASTERISK, IF SO NEEDS PART OF INPUT ADDED
```

```
530 FLAG=0
540 IF RIGHT$(G$,1)="*" THEN FLAG=1:G$=LEFT$(G$,LEN(G$)-1)
550 PRINT G$;" ";
560 IF FLAG=0 THEN 140:REM NO NEED FOR ADDED MATERIAL
570 REM *** NOW USE BALANCE OF INPUT ***
580 X$=" "+MID$(X$,LN+2,LEN(X$))+" "
590 REM **********************
600 REM *** CONJUGATION CHANGES ***
610 REM *** ALSO LOOK FOR 'MY' TO TRIGGER 'MYFLAG' ***
620 LN=LEN(X$)
630 M=0
640 M=M+1
650 IF M=LN THEN 140
660 IF MID$(X$,M,1)=" " THEN 680
670 GOTO 640
680 X=M+1
690 Y=0
700 Y=Y+1
710 IF MID$(X$,(X+Y),1)=" " THEN Q$=MID$(X$,X,Y):GOTO 740
720 IF X+Y>250 THEN 640
730 GOTO 700
740 MN=0
750 MN=MN+1
760 IF Q$="MY" AND K$="" THEN K$=MID$(X$,X+3,LEN(X$)-4):K$=LEFT$(K$,LEN(K$)-1)
770 IF Q$=A$(MN) THEN PRINT B$(MN);" ";:GOTO 640
780 IF MN<KK THEN 750
790 PRINT Q$;" ";
800 GOTO 640
810 REM **********************
820 REM *** TRIGGER WORDS FOUND ***
830 T=INT(RND(1)*3)+1
840 Q$=""
850 IF T=1 THEN Q$=D$(N)
860 IF T=2 THEN Q$=E$(N)
870 IF T=3 THEN Q$=F$(N)
880 IF RIGHT$(Q$,1)<>"*" THEN PRINT Q$:GOTO 140
890 REM *** FALLS THRU NEXT SECTION IF TRIGGER WORD JUDGED UNSUITABLE ***
900 REM ****************************
910 REM *** RANDOM REPLIES/NO TRIGGER
920 IF K$<>"" THEN 1120:REM 'MYFLAG' IS NOT EMPTY, SO GO THERE
930 T=INT(RND(1)*8)+1
940 ON T GOSUB 960,980,1000,1020,1040,1060,1080,1100
950 GOTO 140
960 PRINT "WHAT DOES THAT SUGGEST TO YOU?"
970 RETURN
980 PRINT "I'M NOT SURE I UNDERSTAND THAT FULLY"
990 RETURN
1000 PRINT "PLEASE ELABORATE ON THAT"
```

```
1010 RETURN
1020 PRINT "THAT'S VERY INTERESTING"
1030 RETURN
1040 PRINT "WELL...PLEASE CONTINUE..."
1050 RETURN
1060 PRINT "WHY?"
1070 RETURN
1080 PRINT "AND THEN?"
1090 RETURN
1100 PRINT "I SEE...PLEASE TELL ME MORE ON THAT"
1110 RETURN
1120 REM *** USE 'MYFLAG' ***
1130 T=INT(RND(1)*8)+1
1140 IF T=1 THEN PRINT "TELL ME MORE ABOUT YOUR ";K$
1150 IF T=2 THEN PRINT "EARLIER YOU MENTIONED YOUR ";K$:PRINT "TELL ME A BIT
     MORE ABOUT THAT"
1160 IF T=3 THEN PRINT "DOES THAT HAVE ANYTHING TO DO WITH YOUR ";K$;"?"
1170 IF T=4 THEN PRINT "IS THERE A LINK THERE WITH YOUR ";K$;"?"
1180 IF T=5 THEN PRINT "WHY DON'T WE GO BACK AND DISCUSS YOUR ";K$;" A LITTLE
     MORE?"
1190 IF T=6 THEN PRINT "DOES ANY CONNECTION BETWEEN THAT AND YOUR ";K$;
     " SUGGEST ITSELF?"
1200 IF T=7 THEN PRINT "WOULD YOU PREFER TO TALK ABOUT YOUR ";K$
1210 IF T=8 THEN PRINT "I THINK PERHAPS WORRIES ABOUT YOUR ";K$;" ARE
     BOTHERING YOU"
1220 IF RND(1)>.7 THEN K$=""
1230 GOTO 140
1240 REM ************************
1250 REM *** INITIALIZATION ***
1260 REM ************************
1270 CLS:KEY OFF
1280 REM ************************
1290 DIM A$(14),B$(14):REM CONJUGATIONS
1300 DIM C$(45),D$(45),E$(45),F$(45):REM TRIGGER WORDS AND REPLIES
1310 Z$="":REM TO STOP REPETITIONS
1320 K$="":REM 'MYFLAG'
1330 PRINT:PRINT
1340 PRINT "THE DOCTOR IS IN...PLEASE STAND BY"
1350 KK=0
1360 KK=KK+1
1370 READ A$(KK),B$(KK)
1380 IF B$(KK)="*" THEN 1400
1390 GOTO 1360
1400 K=0
1410 K=K+1
1420 READ C$(K),D$(K),E$(K),F$(K)
1430 IF F$(K)="*" THEN 1450
1440 GOTO 1410
```

```
1450 CLS
1460 PRINT "WELCOME TO ANOTHER SESSION"
1470 PRINT "  WITH THE DOCTOR....."
1480 PRINT
1490 PRINT "HELLO..."
1500 RETURN
1510 REM ************************
1520 REM DATA
1530 REM *** CONJUGATIONS ***
1540 DATA AM,ARE,WAS,WERE,I,YOU,MY,YOUR
1550 DATA "YOU'VE","I'VE",YOUR,MY,ARE,AM
1560 DATA "YOU'RE",I AM,YOURS,MINE,YOU,ME,ME,YOU,*,*
1570 REM *** TRIGGER WORDS/REPLY PHRASES ***
1580 DATA I NEED
1590 DATA WHY DO YOU NEED*
1600 DATA WOULD IT REALLY BE HELPFUL IF YOU GOT*
1610 DATA ARE YOU SURE YOU NEED*
1620 DATA "WHY DON'T YOU"
1630 DATA "DO YOU REALLY THINK I DON'T*"
1640 DATA "PERHAPS EVENTUALLY I WILL*"
1650 DATA "DO YOU REALLY WANT ME TO*"
1660 DATA "WHY CAN'T I"
1670 DATA "DO YOU THINK YOU SHOULD BE ABLE TO*"
1680 DATA "WHY CAN'T YOU*"
1690 DATA "PERHAPS YOU HAVEN'T TRIED HARD ENOUGH"
1700 DATA ARE YOU
1710 DATA "WHY ARE YOU INTERESTED IN WHETHER I AM OR NOT*"
1720 DATA "WOULD YOU PREFER IT IF I WERE NOT*"
1730 DATA "PERHAPS YOU SOMETIMES DREAM I AM*"
1740 DATA "I CAN'T"
1750 DATA "HOW DO YOU KNOW YOU CAN'T*"
1760 DATA "HAVE YOU TRIED?"
1770 DATA "PERHAPS, NOW, YOU CAN*"
1780 DATA I AM
1790 DATA DID YOU COME TO ME BECAUSE YOU ARE*
1800 DATA HOW LONG HAVE YOU BEEN*
1810 DATA "DO YOU THINK IT'S ABSOLUTELY NORMAL TO BE*"
1820 DATA "I'M"
1830 DATA "DO YOU ENJOY BEING*"
1340 DATA "WHY TELL ME YOU'RE*"
1850 DATA "WHY ARE YOU*"
1860 DATA "I WANT"
1870 DATA "WHAT WOULD IT MEAN TO YOU IF YOU GOT*"
1880 DATA "WHY DO YOU WANT*"
1890 DATA "WHAT WOULD IT ADD TO YOUR LIFE IF YOU GOT*"
1900 DATA WHAT
1910 DATA WHY DO YOU ASK?
1920 DATA HOW WOULD AN ANSWER TO THAT HELP YOU?
```

```
1930 DATA WHAT DO YOU THINK
1940 DATA HOW
1950 DATA HOW WOULD YOU SOLVE THAT?
1960 DATA IT WOULD BE BEST TO ANSWER THAT FOR YOURSELF
1970 DATA "WHAT IS IT YOU'RE REALLY ASKING?"
1980 DATA WHO
1990 DATA DO YOU OFTEN THINK ABOUT SUCH QUESTIONS?
2000 DATA "WHAT ANSWER WOULD PUT YOUR MIND AT REST?"
2010 DATA WHO DO YOU THINK*
2020 DATA WHERE
2030 DATA "THAT'S A PRETTY SILLY QUESTION"
2040 DATA "DO YOU REALLY NEED TO KNOW WHERE*"
2050 DATA "WHAT WOULD IT MEAN TO YOU IF I TOLD YOU WHERE*"
2060 DATA WHEN
2070 DATA HOW SHOULD I KNOW WHEN*
2080 DATA THE TIME SHOULD BE DISCUSSED HERE
2090 DATA THINGS HAVE A HABIT OF HAPPENING AT MORE OR LESS THE RIGHT TIME
2100 DATA WHY
2110 DATA "WHY DON'T YOU TELL ME THE REASON WHY*"
2120 DATA WHAT HAVE YOU TOLD ME WHICH WOULD ALLOW ME TO TELL YOU WHY*
2130 DATA DO YOU REALLY NEED TO KNOW WHY*
2140 DATA BECAUSE
2150 DATA IS THAT THE REAL REASON?
2160 DATA WHAT OTHER REASONS COME TO MIND?
2170 DATA WHAT ELSE DOES THAT EXPLAIN?
2180 DATA SORRY
2190 DATA IN WHAT OTHER CIRCUMSTANCES DO YOU APOLOGIZE?
2200 DATA THERE ARE MANY TIMES WHEN NO APOLOGY IS NEEDED
2210 DATA WHAT FEELINGS DO YOU HAVE WHEN YOU APOLOGIZE?
2220 DATA HELLO
2230 DATA "HELLO...IT'S GOOD TO SEE YOU"
2240 DATA "HELLO TO YOU...I'M GLAD YOU COULD DROP BY TODAY"
2250 DATA "HOW ARE YOU...I'M LOOKING FORWARD TO ANOTHER CHAT WITH YOU"
2260 DATA HI
2270 DATA "HI THERE...I'M GLAD TO SEE YOU HERE TODAY"
2280 DATA "HI.  I'M GLAD YOU'VE DROPPED BY...WE'VE GOT LOTS OF TIME TO CHAT"
2290 DATA "HI TO YOU...RELAX NOW, AND LET'S TALK ABOUT YOUR SITUATION"
2300 DATA MAYBE
2310 DATA "YOU SEEM A LITTLE HESITANT"
2320 DATA "THAT'S PRETTY INDECISIVE"
2330 DATA "IN WHAT OTHER SITUATIONS DO YOU SHOW SUCH A TENTATIVE APPROACH?"
2340 DATA NO
2350 DATA "WHY ARE YOU BEING SO NEGATIVE ABOUT IT?"
2360 DATA "ARE YOU SAYING THAT JUST TO BE NEGATIVE?"
2370 DATA "THAT'S PRETTY FORCEFUL...WHAT DOES IT SUGGEST TO YOU?"
2380 DATA ALWAYS
2390 DATA "PLEASE GIVE ME A SPECIFIC EXAMPLE"
2400 DATA "ISN'T 'ALWAYS' A LITTLE STRONG?"
```

```
2410 DATA "WHEN?"
2420 DATA I THINK
2430 DATA "DO YOU DOUBT*"
2440 DATA "DO YOU REALLY THINK SO?"
2450 DATA "BUT YOU ARE NOT SURE*"
2460 DATA FRIEND
2470 DATA "WHY DO YOU BRING UP THE SUBJECT OF FRIENDS?"
2480 DATA "WHAT IS YOUR BEST MEMORY OF A FRIEND?"
2490 DATA "PLEASE TELL ME MORE ABOUT YOUR FRIENDSHIP"
2500 DATA FRIENDS
2510 DATA "IN WHAT WAY DO YOUR FRIENDS' REACTIONS BOTHER YOU?"
2520 DATA "WHAT MADE YOU START TO TALK ABOUT FRIENDS JUST NOW?"
2530 DATA "IN WHAT WAY DO YOUR FRIENDS IMPOSE ON YOU?"
2540 DATA YES
2550 DATA "WHAT LIES BEYOND YOUR CERTAINTY?"
2560 DATA "YOU'RE PRETTY POSITIVE"
2570 DATA "YOU SEEM VERY CERTAIN ABOUT IT"
2580 DATA COMPUTER
2590 DATA "WHAT FEELINGS DO YOU GET SITTING THERE TALKING TO ME LIKE THIS?"
2600 DATA "ARE YOU THINKING ABOUT ME IN PARTICULAR?"
2610 DATA "WHAT ASPECT OF COMPUTERS INTERESTS YOU THE MOST?"
2620 DATA IS IT
2630 DATA "DO YOU THINK IT IS*"
2640 DATA "IN WHAT CIRCUMSTANCES WOULD IT*"
2650 DATA "IT COULD WELL BE THAT*"
2660 DATA IT IS
2670 DATA "WHAT DEGREE OF CERTAINTY WOULD YOU PLACE ON IT BEING*"
2680 DATA "ARE YOU CERTAIN THAT IT'S*"
2690 DATA "WHAT EMOTIONS WOULD YOU FEEL IF I TOLD YOU THAT IT PROBABLY ISN'T*"
2700 DATA CAN YOU
2710 DATA "WHAT MAKES YOU THINK I CAN'T*"
2720 DATA "DON'T YOU THINK THAT I CAN*"
2730 DATA "PERHAPS YOU WOULD LIKE TO BE ABLE TO*"
2740 DATA CAN I
2750 DATA "PERHAPS YOU DON'T WANT TO*"
2760 DATA "DO YOU WANT TO BE ABLE TO*"
2770 DATA "I DOUBT IT"
2780 DATA YOU ARE
2790 DATA "WHY DO YOU THINK I AM*"
2800 DATA "DOES IT PLEASE YOU TO BELIEVE I AM*"
2810 DATA "PERHAPS YOU WOULD LIKE TO BE*"
2820 DATA "YOU'RE"
2830 DATA "WHY DO YOU THINK I AM*"
2840 DATA "DOES IT PLEASE YOU TO BELIEVE I AM*"
2850 DATA "WHY DO YOU SAY I AM*"
2860 DATA "I DON'T"
2870 DATA "DON'T YOU REALLY*"
2880 DATA "WHY DON'T YOU*"
```

```
2890 DATA "DO YOU WANT TO BE ABLE TO*"
2900 DATA I FEEL
2910 DATA "TELL ME MORE ABOUT SUCH FEELINGS"
2920 DATA "DO YOU OFTEN FEEL*"
2930 DATA "DO YOU ENJOY FEELING*"
2940 DATA FEEL
2950 DATA "LET'S EXPLORE THAT STATEMENT A BIT"
2960 DATA "DO YOU OFTEN FEEL LIKE THAT?"
2970 DATA "WHAT EMOTIONS DO SUCH FEELINGS STIR UP IN YOU?"
2980 DATA I HAVE
2990 DATA "WHY TELL ME THAT YOU'VE*"
3000 DATA "IT'S OBVIOUS TO ME THAT YOU HAVE*"
3010 DATA "HOW CAN I HELP YOU WITH*"
3020 DATA I WOULD
3030 DATA "COULD YOU EXPLAIN WHY YOU WOULD*"
3040 DATA "WHO ELSE HAVE YOU TOLD YOU WOULD*"
3050 DATA "HOW SURE ARE YOU THAT YOU WOULD*"
3060 DATA IS THERE
3070 DATA "OF COURSE THERE IS*"
3080 DATA "IT'S LIKELY THAT THERE IS*"
3090 DATA "WOULD YOU LIKE THERE TO BE*"
3100 DATA "MY"
3110 DATA "YOUR*"
3120 DATA "I SEE, YOUR*"
3130 DATA "WHAT DOES IT MEAN TO YOU, THAT YOUR*"
3140 DATA "YOU"
3150 DATA "THIS SESSION IS TO HELP YOU...NOT TO DISCUSS ME!"
3160 DATA "WHAT PROMPTED YOU TO SAY THAT ABOUT ME?"
3170 DATA "REMEMBER, I'M TAKING NOTES ON ALL THIS TO SOLVE YOUR SITUATION"
3180 DATA *,*,*,*
```